Ruby master - Bug #6

sprintf() of %f on Windows(MSVCRT)

05/16/2008 10:13 PM - usa (Usaku NAKAMURA)

Status:	Closed		
Priority:	Normal		
Assignee:	nobu (Nobuyoshi Nakada)		
Target version:	1.9.2		
ruby -v:	ruby 1.9.2dev (2010-02-26) [i386-mingw32]	**Backport:**	

Description

=begin
running test/ruby/test_sprintf.rb(test_float):
<"368934881474191111424"> expected but was
<"368934881474191111000">.

because sprintf() of MSVCRT is not precise.
should use our own dtoa().
=end

Associated revisions

Revision 61562 - 01/02/2018 06:41 AM - shyouhei (Shyouhei Urabe)

fix SEGV touching uninitialized memory

This function can be called from boot_defclass().
No assumption can be made about object internals.

(lldb) run
Process 2386 launched: './miniruby' (x86_64)
Process 2386 stopped

- thread #1: tid = 0x13f3b6, 0x00000001001e0b26 minirubyrb_class_path_cached(klass=4311373720) + 20 at variable.c:321, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x8) frame #0: 0x00000001001e0b26 minirubyrb_class_path_cached(klass=4311373720) + 20 at variable.c:321 318 VALUE 319 rb_class_path_cached(VALUE klass) 320 { -> 321 st_table *ivtbl = RCLASS_IV_TBL(klass); 322 st_data_t n; 323 324 if (!ivtbl) return Qnil; (lldb) bt
- thread #1: tid = 0x13f3b6, 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variable.c:321, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x8)
  - frame #0: 0x00000001001e0b26 minirubyrb_class_path_cached(klass=4311373720) + 20 at variable.c:321 frame #1: 0x000000010009cbd0 minirubyrb_raw_obj_info(buff="0x0000000100fa5798 [2] T_CLASS", buff_size=256, obj=4311373720) + 1393 at gc.c:9341 frame #2: 0x000000010009cf16 minirubyobj_info(obj=4311373720) + 98 at gc.c:9423 frame #3: 0x000000010008ca87 minirubynewobj_init(klass=0, flags=66, v1=0, v2=0, v3=0, wb_protected=1, objspace=0x00000001007cf280, obj=4311373720) + 338 at gc.c:1887 frame #4: 0x000000010008cd51 minirubynewobj_of(klass=0, flags=66, v1=0, v2=0, v3=0, wb_protected=1) + 171 at gc.c:1970 frame #5: 0x000000010008ce1b minirubyrb_wb_protected_newobj_of(klass=0, flags=66) + 54 at gc.c:1990 frame #6: 0x0000000100027563 minirubyclass_alloc(flags=2, klass=0) + 46 at class.c:165 frame #7: 0x000000010002761a minirubyrb_class_boot(super=0) + 35 at class.c:203 frame #8: 0x0000000100028612 minirubyboot_defclass(name="BasicObject", super=0) + 28 at class.c:537 frame #9: 0x000000010002868b minirubyInit_class_hierarchy + 26 at class.c:548 frame #10: 0x00000001000efe69 minirubyInitVM_Object + 9 at object.c:3892 frame #11: 0x00000001000f138e minirubyInit_Object + 57 at object.c:4122 frame #12: 0x00000001000a59bd minirubyrb_call_inits + 29 at inits.c:23 frame #13: 0x000000010007af30 minirubyruby_setup + 229 at eval.c:61 frame #14: 0x000000010007af7e minirubyruby_init + 13 at eval.c:78 frame #15: 0x0000000100000c58 minirubymain(argc=2, argv=0x00007fff5fbfdbf0) + 88 at main.c:41 frame #16: 0x00007fff88eda5ad libdyld.dylib`start + 1 (lldb)

Revision 61562 - 01/02/2018 06:41 AM - shyouhei (Shyouhei Urabe)

fix SEGV touching uninitialized memory

This function can be called from boot_defclass().
No assumption can be made about object internals.

(lldb) run
Process 2386 launched: './miniruby' (x86_64)
Process 2386 stopped

- thread #1: tid = 0x13f3b6, 0x00000001001e0b26 minirubyrb_class_path_cached(klass=4311373720) + 20 at variable.c:321, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x8) frame #0: 0x00000001001e0b26 minirubyrb_class_path_cached(klass=4311373720) + 20 at variable.c:321 318 VALUE 319 rb_class_path_cached(VALUE klass) 320 { -> 321

st_table *ivtbl = RCLASS_IV_TBL(klass); 322 st_data_t n; 323 324 if (!ivtbl) return Qnil; (lldb) bt

- thread #1: tid = 0x13f3b6, 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variable.c:321, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x8)
  - frame #0: 0x00000001001e0b26 minirubyrb_class_path_cached(klass=4311373720) + 20 at variable.c:321 frame #1: 0x000000010009cbd0 minirubyrb_raw_obj_info(buff="0x0000000100fa5798 [2] T_CLASS", buff_size=256, obj=4311373720) + 1393 at gc.c:9341 frame #2: 0x000000010009cf16 minirubyobj_info(obj=4311373720) + 98 at gc.c:9423 frame #3: 0x000000010008ca87 minirubynewobj_init(klass=0, flags=66, v1=0, v2=0, v3=0, wb_protected=1, objspace=0x00000001007cf280, obj=4311373720) + 338 at gc.c:1887 frame #4: 0x000000010008cd51 minirubynewobj_of(klass=0, flags=66, v1=0, v2=0, v3=0, wb_protected=1) + 171 at gc.c:1970 frame #5: 0x000000010008ce1b minirubyrb_wb_protected_newobj_of(klass=0, flags=66) + 54 at gc.c:1990 frame #6: 0x0000000100027563 minirubyclass_alloc(flags=2, klass=0) + 46 at class.c:165 frame #7: 0x000000010002761a minirubyrb_class_boot(super=0) + 35 at class.c:203 frame #8: 0x0000000100028612 minirubyboot_defclass(name="BasicObject", super=0) + 28 at class.c:537 frame #9: 0x000000010002868b minirubyInit_class_hierarchy + 26 at class.c:548 frame #10: 0x00000001000efe69 minirubyInitVM_Object + 9 at object.c:3892 frame #11: 0x00000001000f138e minirubyInit_Object + 57 at object.c:4122 frame #12: 0x00000001000a59bd minirubyrb_call_inits + 29 at inits.c:23 frame #13: 0x000000010007af30 minirubyruby_setup + 229 at eval.c:61 frame #14: 0x000000010007af7e minirubyruby_init + 13 at eval.c:78 frame #15: 0x0000000100000c58 minirubymain(argc=2, argv=0x00007fff5fbfdbf0) + 88 at main.c:41 frame #16: 0x00007fff88eda5ad libdyld.dylib`start + 1 (lldb)

Revision 61563 - 01/02/2018 06:41 AM - shyouhei (Shyouhei Urabe)

fix SEGV touching uninitialized memory

This function can be called from InitVM_Object().
No assumption can be made about object internals.

(lldb) run
Process 10675 launched: './miniruby' (x86_64)
Process 10675 stopped

- thread #1: tid = 0x14252c, 0x00000001000bdda9 minirubyrb_raw_obj_info(buff="0x0000000100fc1588 [0] T_IMEMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0) frame #0: 0x00000001000bdda9 minirubyrb_raw_obj_info(buff="0x0000000100fc1588 [0] T_IMEMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383 9380 const rb_method_entry_t *me = &RANY(obj)->as.imemo.ment; 9381 snprintf(buff, buff_size, "%s (called_id: %s, type: %s, alias: %d, owner: %s, defined_class: %s)", buff, 9382 rb_id2name(me->called_id), -> 9383 method_type_name(me->def->type), 9384 me->def->alias_count, 9385 obj_info(me->owner), 9386 obj_info(me->defined_class)); (lldb) p *me (rb_method_entry_t) $0 = { flags = 24602 defined_class = 4311488400 def = 0x0000000000000000 called_id = 3057 owner = 4311488400 } (lldb) bt
- thread #1: tid = 0x14252c, 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0] T_IMEMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
  - frame #0: 0x00000001000bdda9 minirubyrb_raw_obj_info(buff="0x0000000100fc1588 [0] T_IMEMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383 frame #1: 0x00000001000b7cbf minirubyobj_info(obj=4311487880) + 95 at gc.c:9423 frame #2: 0x00000001000c16a8 minirubynewobj_init(klass=4311488400, flags=24602, v1=0, v2=3057, v3=4311488400, wb_protected=1, objspace=0x00000001007ee280, obj=4311487880) + 424 at gc.c:1887 frame #3: 0x00000001000b4529 minirubynewobj_of(klass=4311488400, flags=24602, v1=0, v2=3057, v3=4311488400, wb_protected=1) + 217 at gc.c:1970 frame #4: 0x00000001000b46ab minirubyrb_imemo_new(type=imemo_ment, v1=0, v2=3057, v3=4311488400, v0=4311488400) + 75 at gc.c:2017 frame #5: 0x00000001002773b4 minirubyrb_method_entry_alloc(called_id=3057, owner=4311488400, defined_class=4311488400, def=0x0000000000000000) + 52 at vm_method.c:368 frame #6: 0x0000000100277307 minirubyrb_method_entry_create(called_id=3057, klass=4311488400, visi=METHOD_VISI_PRIVATE, def=0x0000000000000000) + 71 at vm_method.c:389 frame #7: 0x00000001002784c7 minirubyrb_method_entry_make(klass=4311488400, mid=3057, defined_class=4311488400, visi=METHOD_VISI_PRIVATE, type=VM_METHOD_TYPE_CFUNC, def=0x0000000000000000, original_id=3057, opts=0x00007fff5fbfd9e8) + 1207 at vm_method.c:594 frame #8: 0x00000001002770f9 minirubyrb_add_method(klass=4311488400, mid=3057, type=VM_METHOD_TYPE_CFUNC, opts=0x00007fff5fbfd9e8, visi=METHOD_VISI_PRIVATE) + 73 at vm_method.c:650 frame #9: 0x000000010027708a minirubyrb_add_method_cfunc(klass=4311488400, mid=3057, func=(minirubyrb_obj_dummy at object.c:1125), argc=0, visi=METHOD_VISI_PRIVATE) + 138 at vm_method.c:137 frame #10: 0x00000001000391e4 minirubyrb_define_private_method(klass=4311488400, name="initialize", func=(minirubyrb_obj_dummy at object.c:1125), argc=0) + 68 at class.c:1529 frame #11: 0x000000010013f5bf minirubyInitVM_Object + 47 at object.c:3905 frame #12: 0x0000000100142ffd minirubyInit_Object + 61 at object.c:4122 frame #13: 0x00000001000d4edd minirubyrb_call_inits + 29 at inits.c:23 frame #14: 0x000000010009fe66 minirubyruby_setup + 198 at eval.c:61 frame #15: 0x000000010009febd minirubyruby_init + 13 at eval.c:78 frame #16: 0x0000000100000a4d minirubymain(argc=2, argv=0x00007fff5fbfdbf0) + 93 at main.c:41 frame #17: 0x00007fff88eda5ad libdyld.dylibstart + 1 (lldb)

Revision 61563 - 01/02/2018 06:41 AM - shyouhei (Shyouhei Urabe)

fix SEGV touching uninitialized memory

This function can be called from InitVM_Object().
No assumption can be made about object internals.

(lldb) run
Process 10675 launched: './miniruby' (x86_64)
Process 10675 stopped

- thread #1: tid = 0x14252c, 0x00000001000bdda9 minirubyrb_raw_obj_info(buff="0x0000000100fc1588 [0] T_IMEMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0) frame #0: 0x00000001000bdda9 minirubyrb_raw_obj_info(buff="0x0000000100fc1588 [0] T_IMEMO ment", buff_size=256, obj=4311487880) + 2489 at

gc.c:9383 9380 const rb_method_entry_t *me = &RANY(obj)->as.imemo.ment; 9381 snprintf(buff, buff_size, "%s (called_id: %s, type: %s, alias: %d, owner: %s, defined_class: %s)", buff, 9382 rb_id2name(me->called_id), -> 9383 method_type_name(me->def->type), 9384 me->def->alias_count, 9385 obj_info(me->owner), 9386 obj_info(me->defined_class)); (lldb) p *me (rb_method_entry_t) $0 = { flags = 24602 defined_class = 4311488400 def = 0x0000000000000000 called_id = 3057 owner = 4311488400 } (lldb) bt

- thread #1: tid = 0x14252c, 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0] T_IMEMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
  - frame #0: 0x00000001000bdda9 minirubyrb_raw_obj_info(buff="0x0000000100fc1588 [0] T_IMEMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383 frame #1: 0x00000001000b7cbf minirubyobj_info(obj=4311487880) + 95 at gc.c:9423 frame #2: 0x00000001000c16a8 minirubynewobj_init(klass=4311488400, flags=24602, v1=0, v2=3057, v3=4311488400, wb_protected=1, objspace=0x00000001007ee280, obj=4311487880) + 424 at gc.c:1887 frame #3: 0x00000001000b4529 minirubynewobj_of(klass=4311488400, flags=24602, v1=0, v2=3057, v3=4311488400, wb_protected=1) + 217 at gc.c:1970 frame #4: 0x00000001000b46ab minirubyrb_imemo_new(type=imemo_ment, v1=0, v2=3057, v3=4311488400, v0=4311488400) + 75 at gc.c:2017 frame #5: 0x00000001002773b4 minirubyrb_method_entry_alloc(called_id=3057, owner=4311488400, defined_class=4311488400, def=0x0000000000000000) + 52 at vm_method.c:368 frame #6: 0x0000000100277307 minirubyrb_method_entry_create(called_id=3057, klass=4311488400, visi=METHOD_VISI_PRIVATE, def=0x0000000000000000) + 71 at vm_method.c:389 frame #7: 0x00000001002784c7 minirubyrb_method_entry_make(klass=4311488400, mid=3057, defined_class=4311488400, visi=METHOD_VISI_PRIVATE, type=VM_METHOD_TYPE_CFUNC, def=0x0000000000000000, original_id=3057, opts=0x00007fff5fbfd9e8) + 1207 at vm_method.c:594 frame #8: 0x00000001002770f9 minirubyrb_add_method(klass=4311488400, mid=3057, type=VM_METHOD_TYPE_CFUNC, opts=0x00007fff5fbfd9e8, visi=METHOD_VISI_PRIVATE) + 73 at vm_method.c:650 frame #9: 0x000000010027708a minirubyrb_add_method_cfunc(klass=4311488400, mid=3057, func=(minirubyrb_obj_dummy at object.c:1125), argc=0, visi=METHOD_VISI_PRIVATE) + 138 at vm_method.c:137 frame #10: 0x00000001000391e4 minirubyrb_define_private_method(klass=4311488400, name="initialize", func=(minirubyrb_obj_dummy at object.c:1125), argc=0) + 68 at class.c:1529 frame #11: 0x000000010013f5bf minirubyInitVM_Object + 47 at object.c:3905 frame #12: 0x0000000100142ffd minirubyInit_Object + 61 at object.c:4122 frame #13: 0x00000001000d4edd minirubyrb_call_inits + 29 at inits.c:23 frame #14: 0x000000010009fe66 minirubyruby_setup + 198 at eval.c:61 frame #15: 0x000000010009febd minirubyruby_init + 13 at eval.c:78 frame #16: 0x0000000100000a4d minirubymain(argc=2, argv=0x00007fff5fbfdbf0) + 93 at main.c:41 frame #17: 0x00007fff88eda5ad libdyld.dylibstart + 1 (lldb)

Revision 61564 - 01/02/2018 06:42 AM - shyouhei (Shyouhei Urabe)

fix SEGV touching uninitialized memory

This function can be called from Init_VM().
No assumption can be made about object internals.

(lldb) run
Process 15734 launched: './miniruby' (x86_64)
Process 15734 stopped

- thread #1: tid = 0x1441d4, 0x00000001000bdfcb minirubyrb_raw_iseq_info(buff="0x0000000100f61f48 [0] T_IMEMO iseq", buff_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x50) frame #0: 0x00000001000bdfcb minirubyrb_raw_iseq_info(buff="0x0000000100f61f48 [0] T_IMEMO iseq", buff_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273 9270 static void 9271 rb_raw_iseq_info(char *buff, const int buff_size, const rb_iseq_t *iseq) 9272 { -> 9273 if (iseq->body->location.label) { 9274 VALUE path = rb_iseq_path(iseq); 9275 snprintf(buff, buff_size, "%s %s@%s:%d", buff, 9276 RSTRING_PTR(iseq->body->location.label), (lldb) p *iseq (rb_iseq_t) $0 = { flags = 28698 reserved1 = 0 body = 0x0000000000000000 aux = { compile_data = 0x0000000000000000 loader = (obj = 0, index = 0) trace_events = 0 } } (lldb) bt
- thread #1: tid = 0x1441d4, 0x00000001000bdfcb miniruby`rb_raw_iseq_info(buff="0x0000000100f61f48 [0] T_IMEMO iseq", buff_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x50)
  - frame #0: 0x00000001000bdfcb minirubyrb_raw_iseq_info(buff="0x0000000100f61f48 [0] T_IMEMO iseq", buff_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273 frame #1: 0x00000001000bde72 minirubyrb_raw_obj_info(buff="0x0000000100f61f48 [0] T_IMEMO iseq", buff_size=256, obj=4311097160) + 2786 at gc.c:9396 frame #2: 0x00000001000b7c5f minirubyobj_info(obj=4311097160) + 95 at gc.c:9428 frame #3: 0x00000001000c16a8 minirubynewobj_init(klass=0, flags=28698, v1=0, v2=0, v3=0, wb_protected=1, objspace=0x00000001007ee280, obj=4311097160) + 424 at gc.c:1887 frame #4: 0x00000001000b44c9 minirubynewobj_of(klass=0, flags=28698, v1=0, v2=0, v3=0, wb_protected=1) + 217 at gc.c:1970 frame #5: 0x00000001000b464b minirubyrb_imemo_new(type=imemo_iseq, v1=0, v2=0, v3=0, v0=0) + 75 at gc.c:2017 frame #6: 0x00000001000fd914 minirubyiseq_imemo_alloc + 36 at iseq.h:156 frame #7: 0x00000001000f6e1d minirubyiseq_alloc + 13 at iseq.c:211 frame #8: 0x00000001000f6bf8 minirubyrb_iseq_new_with_opt(node=0x0000000000000000, name=4311097200, path=4311097200, realpath=8, first_lineno=1, parent=0x0000000000000000, type=ISEQ_TYPE_TOP, option=0x0000000100335c30) + 56 at iseq.c:519 frame #9: 0x00000001000f6bb6 minirubyrb_iseq_new(node=0x0000000000000000, name=4311097200, path=4311097200, realpath=8, parent=0x0000000000000000, type=ISEQ_TYPE_TOP) + 86 at iseq.c:480 frame #10: 0x0000000100284bb0 minirubyInit_VM + 1040 at vm.c:3022 frame #11: 0x00000001000d4f7d minirubyrb_call_inits + 189 at inits.c:55 frame #12: 0x000000010009fe06 minirubyruby_setup + 198 at eval.c:61 frame #13: 0x000000010009fe5d minirubyruby_init + 13 at eval.c:78 frame #14: 0x00000001000009ed minirubymain(argc=2, argv=0x00007fff5fbfdbf0) + 93 at main.c:41 frame #15: 0x00007fff88eda5ad libdyld.dylibstart + 1 (lldb)

Revision 61564 - 01/02/2018 06:42 AM - shyouhei (Shyouhei Urabe)

fix SEGV touching uninitialized memory

This function can be called from Init_VM().
No assumption can be made about object internals.

(lldb) run
Process 15734 launched: './miniruby' (x86_64)
Process 15734 stopped

- thread #1: tid = 0x1441d4, 0x00000001000bdfcb minirubyrb_raw_iseq_info(buff="0x0000000100f61f48 [0] T_IMEMO iseq", buff_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x50) frame #0: 0x00000001000bdfcb minirubyrb_raw_iseq_info(buff="0x0000000100f61f48 [0] T_IMEMO iseq", buff_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273 9270 static void 9271 rb_raw_iseq_info(char *buff, const int buff_size, const rb_iseq_t *iseq) 9272 { -> 9273 if (iseq->body->location.label) { 9274 VALUE path = rb_iseq_path(iseq); 9275 snprintf(buff, buff_size, "%s %s@%s:%d", buff, 9276 RSTRING_PTR(iseq->body->location.label), (lldb) p *iseq (rb_iseq_t) $0 = { flags = 28698 reserved1 = 0 body = 0x0000000000000000 aux = { compile_data = 0x0000000000000000 loader = (obj = 0, index = 0) trace_events = 0 } } (lldb) bt
- thread #1: tid = 0x1441d4, 0x00000001000bdfcb miniruby`rb_raw_iseq_info(buff="0x0000000100f61f48 [0] T_IMEMO iseq", buff_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x50)
  - frame #0: 0x00000001000bdfcb minirubyrb_raw_iseq_info(buff="0x0000000100f61f48 [0] T_IMEMO iseq", buff_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273 frame #1: 0x00000001000bde72 minirubyrb_raw_obj_info(buff="0x0000000100f61f48 [0] T_IMEMO iseq", buff_size=256, obj=4311097160) + 2786 at gc.c:9396 frame #2: 0x00000001000b7c5f minirubyobj_info(obj=4311097160) + 95 at gc.c:9428 frame #3: 0x00000001000c16a8 minirubynewobj_init(klass=0, flags=28698, v1=0, v2=0, v3=0, wb_protected=1, objspace=0x00000001007ee280, obj=4311097160) + 424 at gc.c:1887 frame #4: 0x00000001000b44c9 minirubynewobj_of(klass=0, flags=28698, v1=0, v2=0, v3=0, wb_protected=1) + 217 at gc.c:1970 frame #5: 0x00000001000b464b minirubyrb_imemo_new(type=imemo_iseq, v1=0, v2=0, v3=0, v0=0) + 75 at gc.c:2017 frame #6: 0x00000001000fd914 minirubyiseq_imemo_alloc + 36 at iseq.h:156 frame #7: 0x00000001000f6e1d minirubyiseq_alloc + 13 at iseq.c:211 frame #8: 0x00000001000f6bf8 minirubyrb_iseq_new_with_opt(node=0x0000000000000000, name=4311097200, path=4311097200, realpath=8, first_lineno=1, parent=0x0000000000000000, type=ISEQ_TYPE_TOP, option=0x0000000100335c30) + 56 at iseq.c:519 frame #9: 0x00000001000f6bb6 minirubyrb_iseq_new(node=0x0000000000000000, name=4311097200, path=4311097200, realpath=8, parent=0x0000000000000000, type=ISEQ_TYPE_TOP) + 86 at iseq.c:480 frame #10: 0x0000000100284bb0 minirubyInit_VM + 1040 at vm.c:3022 frame #11: 0x00000001000d4f7d minirubyrb_call_inits + 189 at inits.c:55 frame #12: 0x000000010009fe06 minirubyruby_setup + 198 at eval.c:61 frame #13: 0x000000010009fe5d minirubyruby_init + 13 at eval.c:78 frame #14: 0x00000001000009ed minirubymain(argc=2, argv=0x00007fff5fbfdbf0) + 93 at main.c:41 frame #15: 0x00007fff88eda5ad libdyld.dylibstart + 1 (lldb)

Revision 61565 - 01/02/2018 06:42 AM - shyouhei (Shyouhei Urabe)

fix SEGV touching uninitialized memory

This function can be called from rb_data_typed_object_zalloc().
No assumption can be made about object internals.

(lldb) run
Process 22135 launched: './miniruby' (x86_64)
Process 22135 stopped

- thread #1: tid = 0x14a3af, 0x000000010008ac8a minirubyvm_block_type(block=0x0000000000000000) + 12 at vm_core.h:1364, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x18) frame #0: 0x000000010008ac8a minirubyvm_block_type(block=0x0000000000000000) + 12 at vm_core.h:1364 1361 break; 1362 } 1363 #endif -> 1364 return block->type; 1365 } 1366 1367 static inline void (lldb) bt
- thread #1: tid = 0x14a3af, 0x000000010008ac8a miniruby`vm_block_type(block=0x0000000000000000) + 12 at vm_core.h:1364, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x18)
  - frame #0: 0x000000010008ac8a minirubyvm_block_type(block=0x0000000000000000) + 12 at vm_core.h:1364 frame #1: 0x000000010008acdb minirubyvm_block_iseq(block=0x0000000000000000) + 24 at vm_core.h:1399 frame #2: 0x000000010008acc1 minirubyvm_proc_iseq(procval=4310866360) + 32 at vm_core.h:1387 frame #3: 0x000000010009cbed minirubyrb_raw_obj_info(buff="0x0000000100f299b8 [0] proc (Proc)", buff_size=256, obj=4310866360) + 1513 at gc.c:9349 frame #4: 0x000000010009cf01 minirubyobj_info(obj=4310866360) + 98 at gc.c:9428 frame #5: 0x000000010008ca1b minirubynewobj_init(klass=4311027960, flags=12, v1=4298186080, v2=1, v3=0, wb_protected=32, objspace=0x00000001007cf280, obj=4310866360) + 338 at gc.c:1887 frame #6: 0x000000010008cce5 minirubynewobj_of(klass=4311027960, flags=12, v1=4298186080, v2=1, v3=0, wb_protected=32) + 171 at gc.c:1970 frame #7: 0x000000010008d01d minirubyrb_data_typed_object_wrap(klass=4311027960, datap=0x0000000000000000, type=0x0000000100311d60) + 133 at gc.c:2062 frame #8: 0x000000010008d04e minirubyrb_data_typed_object_zalloc(klass=4311027960, size=40, type=0x0000000100311d60) + 42 at gc.c:2073 frame #9: 0x000000010011b459 minirubyrb_proc_alloc(klass=4311027960) + 36 at proc.c:113 frame #10: 0x0000000100204d8e minirubyvm_proc_create_from_captured(klass=4311027960, captured=0x00000001025003f8, block_type=block_type_iseq, is_from_method='\0', is_lambda='\x01') + 44 at vm.c:814 frame #11: 0x00000001002050d8 minirubyrb_vm_make_proc_lambda(ec=0x00000001007cf548, captured=0x00000001025003f8, klass=4311027960, is_lambda='\x01') + 134 at vm.c:892 frame #12: 0x000000010011c0d2 minirubyproc_new(klass=4311027960, is_lambda='\x01') + 445 at proc.c:752 frame #13: 0x000000010011c154 minirubyrb_block_lambda + 27 at proc.c:808 frame #14: 0x00000001001ee7e3 minirubycall_cfunc_0(func=(minirubyrb_block_lambda at proc.c:807), recv=4310991600, argc=0, argv=0x0000000102400480) + 41 at vm_insnhelper.c:1729 frame #15: 0x00000001001ef2c3 minirubyvm_call_cfunc_with_frame(ec=0x00000001007cf548, reg_cfp=0x00000001025003e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x000000010253e0f0) + 386 at vm_insnhelper.c:1918 frame #16: 0x00000001001ef412 minirubyvm_call_cfunc(ec=0x00000001007cf548, reg_cfp=0x00000001025003e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x000000010253e0f0) + 149 at vm_insnhelper.c:1934 frame #17: 0x00000001001f0655 minirubyvm_call_method_each_type(ec=0x00000001007cf548, cfp=0x00000001025003e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x000000010253e0f0) + 239 at vm_insnhelper.c:2232 frame #18: 0x00000001001f0ce0 minirubyvm_call_method(ec=0x00000001007cf548, cfp=0x00000001025003e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x000000010253e0f0) + 117 at vm_insnhelper.c:2355 frame #19: 0x00000001001f0eb6 minirubyvm_call_general(ec=0x00000001007cf548, reg_cfp=0x00000001025003e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x000000010253e0f0) + 59 at vm_insnhelper.c:2398 frame #20: 0x00000001001f6e61 minirubyvm_exec_core(ec=0x00000001007cf548, initial=0) + 7480 at insns.def:850 frame #21: 0x0000000100207995 minirubyvm_exec(ec=0x00000001007cf548) + 230 at vm.c:1771 frame #22: 0x0000000100208647 minirubyrb_iseq_eval_main(iseq=0x0000000100f29fd0) + 52 at vm.c:2019 frame #23: 0x000000010007b750 minirubyruby_exec_internal(n=0x0000000100f29fd0) + 297 at eval.c:246 frame #24: 0x000000010007b876 minirubyruby_exec_node(n=0x0000000100f29fd0) + 36 at eval.c:310 frame #25: 0x000000010007b849 minirubyruby_run_node(n=0x0000000100f29fd0) + 62 at eval.c:302 frame #26: 0x0000000100000c05 minirubymain(argc=2,

argv=0x00007fff5fbfdbf0) + 113 at main.c:42 frame #27: 0x00007fff88eda5ad libdyld.dylib`start + 1 (lldb)

Revision 61565 - 01/02/2018 06:42 AM - shyouhei (Shyouhei Urabe)

fix SEGV touching uninitialized memory

This function can be called from rb_data_typed_object_zalloc().
No assumption can be made about object internals.

(lldb) run
Process 22135 launched: './miniruby' (x86_64)
Process 22135 stopped

- thread #1: tid = 0x14a3af, 0x000000010008ac8a minirubyvm_block_type(block=0x0000000000000000) + 12 at vm_core.h:1364, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x18) frame #0: 0x000000010008ac8a minirubyvm_block_type(block=0x0000000000000000) + 12 at vm_core.h:1364 1361 break; 1362 } 1363 #endif -> 1364 return block->type; 1365 } 1366 1367 static inline void (lldb) bt
- thread #1: tid = 0x14a3af, 0x000000010008ac8a miniruby`vm_block_type(block=0x0000000000000000) + 12 at vm_core.h:1364, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x18)
  - frame #0: 0x000000010008ac8a minirubyvm_block_type(block=0x0000000000000000) + 12 at vm_core.h:1364 frame #1: 0x000000010008acdb minirubyvm_block_iseq(block=0x0000000000000000) + 24 at vm_core.h:1399 frame #2: 0x000000010008acc1 minirubyvm_proc_iseq(procval=4310866360) + 32 at vm_core.h:1387 frame #3: 0x000000010009cbed minirubyrb_raw_obj_info(buff="0x0000000100f299b8 [0] proc (Proc)", buff_size=256, obj=4310866360) + 1513 at gc.c:9349 frame #4: 0x000000010009cf01 minirubyobj_info(obj=4310866360) + 98 at gc.c:9428 frame #5: 0x000000010008ca1b minirubynewobj_init(klass=4311027960, flags=12, v1=4298186080, v2=1, v3=0, wb_protected=32, objspace=0x00000001007cf280, obj=4310866360) + 338 at gc.c:1887 frame #6: 0x000000010008cce5 minirubynewobj_of(klass=4311027960, flags=12, v1=4298186080, v2=1, v3=0, wb_protected=32) + 171 at gc.c:1970 frame #7: 0x000000010008d01d minirubyrb_data_typed_object_wrap(klass=4311027960, datap=0x0000000000000000, type=0x0000000100311d60) + 133 at gc.c:2062 frame #8: 0x000000010008d04e minirubyrb_data_typed_object_zalloc(klass=4311027960, size=40, type=0x0000000100311d60) + 42 at gc.c:2073 frame #9: 0x000000010011b459 minirubyrb_proc_alloc(klass=4311027960) + 36 at proc.c:113 frame #10: 0x0000000100204d8e minirubyvm_proc_create_from_captured(klass=4311027960, captured=0x0000000001025003f8, block_type=block_type_iseq, is_from_method='\0', is_lambda='\x01') + 44 at vm.c:814 frame #11: 0x00000001002050d8 minirubyrb_vm_make_proc_lambda(ec=0x00000001007cf548, captured=0x0000000001025003f8, klass=4311027960, is_lambda='\x01') + 134 at vm.c:892 frame #12: 0x000000010011c0d2 minirubyproc_new(klass=4311027960, is_lambda='\x01') + 445 at proc.c:752 frame #13: 0x000000010011c154 minirubyrb_block_lambda + 27 at proc.c:808 frame #14: 0x00000001001ee7e3 minirubycall_cfunc_0(func=(minirubyrb_block_lambda at proc.c:807), recv=4310991600, argc=0, argv=0x0000000102400480) + 41 at vm_insnhelper.c:1729 frame #15: 0x00000001001ef2c3 minirubyvm_call_cfunc_with_frame(ec=0x00000001007cf548, reg_cfp=0x0000000001025003e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x000000010253e0f0) + 386 at vm_insnhelper.c:1918 frame #16: 0x00000001001ef412 minirubyvm_call_cfunc(ec=0x00000001007cf548, reg_cfp=0x0000000001025003e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x000000010253e0f0) + 149 at vm_insnhelper.c:1934 frame #17: 0x00000001001f0655 minirubyvm_call_method_each_type(ec=0x00000001007cf548, cfp=0x0000000001025003e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x000000010253e0f0) + 239 at vm_insnhelper.c:2232 frame #18: 0x00000001001f0ce0 minirubyvm_call_method(ec=0x00000001007cf548, cfp=0x0000000001025003e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x000000010253e0f0) + 117 at vm_insnhelper.c:2355 frame #19: 0x00000001001f0eb6 minirubyvm_call_general(ec=0x00000001007cf548, reg_cfp=0x0000000001025003e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x000000010253e0f0) + 59 at vm_insnhelper.c:2398 frame #20: 0x00000001001f6e61 minirubyvm_exec_core(ec=0x00000001007cf548, initial=0) + 7480 at insns.def:850 frame #21: 0x0000000100207995 minirubyvm_exec(ec=0x00000001007cf548) + 230 at vm.c:1771 frame #22: 0x0000000100208647 minirubyrb_iseq_eval_main(iseq=0x0000000100f29fd0) + 52 at vm.c:2019 frame #23: 0x000000010007b750 minirubyruby_exec_internal(n=0x0000000100f29fd0) + 297 at eval.c:246 frame #24: 0x000000010007b876 minirubyruby_exec_node(n=0x0000000100f29fd0) + 36 at eval.c:310 frame #25: 0x000000010007b849 minirubyruby_run_node(n=0x0000000100f29fd0) + 62 at eval.c:302 frame #26: 0x0000000100000c05 minirubymain(argc=2, argv=0x00007fff5fbfdbf0) + 113 at main.c:42 frame #27: 0x00007fff88eda5ad libdyld.dylib`start + 1 (lldb)

Revision 61566 - 01/02/2018 06:42 AM - shyouhei (Shyouhei Urabe)

fix SEGV touching uninitialized local variable

This imemo_name is used uninitialized because the switch
above does not cover all possible imemo types.

(lldb) run
Process 26068 launched: './miniruby' (x86_64)
Process 26068 stopped

- thread #1: tid = 0x14ba96, 0x00007fff8a402132 libsystem_c.dylibstrlen + 18, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0xffffffffffffffff0) frame #0: 0x00007fff8a402132 libsystem_c.dylibstrlen + 18 libsystem_c.dylib`strlen: -> 0x7fff8a402132 <+18>: pcmpeqb (%rdi), %xmm0 0x7fff8a402136 <+22>: pmovmskb %xmm0, %esi 0x7fff8a40213a <+26>: andq $0xf, %rcx 0x7fff8a40213e <+30>: orq $-0x1, %rax (lldb) bt
- thread #1: tid = 0x14ba96, 0x00007fff8a402132 libsystem_c.dylib`strlen + 18, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0xffffffffffffffff0)
  - frame #0: 0x00007fff8a402132 libsystem_c.dylibstrlen + 18 frame #1: 0x00000001001f1531 minirubyBSD_vfprintf(fp=0x00007fff5fbfc9e0, fmt0="%s %s", ap=0x00007fff5fbfcbf0) + 5873 at vsnprintf.c:1026 frame #2: 0x00000001001ef213 minirubyruby_do_vsnprintf(str="0x0000000100f46450 [0] T_IMEMO", n=256, fmt="%s %s", ap=0x00007fff5fbfcbf0) + 131 at sprintf.c:1285 frame #3: 0x00000001001ef3ea minirubyruby_snprintf(str="0x0000000100f46450 [0] T_IMEMO", n=256, fmt="%s %s") +

426 at sprintf.c:1300 frame #4: 0x00000001000bdc61 minirubyrb_raw_obj_info(buff="0x0000000100f46450 [0] T_IMEMO", buff_size=256, obj=4310983760) + 2353 at gc.c:9376 frame #5: 0x00000001000b7bff minirubyobj_info(obj=4310983760) + 95 at gc.c:9428 frame #6: 0x00000001000c1658 minirubynewobj_init(klass=0, flags=36890, v1=0, v2=4303040512, v3=4310983800, wb_protected=1, objspace=0x00000001007ee280, obj=4310983760) + 424 at gc.c:1887 frame #7: 0x00000001000b4469 minirubynewobj_of(klass=0, flags=36890, v1=0, v2=4303040512, v3=4310983800, wb_protected=1) + 217 at gc.c:1970 frame #8: 0x00000001000b45eb minirubyrb_imemo_new(type=imemo_ast, v1=0, v2=4303040512, v3=4310983800, v0=0) + 75 at gc.c:2017 frame #9: 0x000000010011daed minirubyrb_ast_new + 61 at node.c:1146 frame #10: 0x0000000100160e15 minirubyrb_parser_compile_file_path(vparser=4310984400, fname=4310984960, file=4310984080, start=1) + 53 at parse.y:5776 frame #11: 0x00000001001e18ea minirubyload_file_internal(argp_v=140734799795024) + 1834 at ruby.c:1907 frame #12: 0x00000001000a1bb5 minirubyrb_ensure(b_proc=(minirubyload_file_internal at ruby.c:1795), data1=140734799795024, e_proc=(minirubyrestore_load_file at ruby.c:2007), data2=140734799795024) + 245 at eval.c:1037 frame #13: 0x00000001001df4a4 minirubyload_file(parser=4310984400, fname=4310984960, f=4310984080, script=1, opt=0x00007fff5fbfda28) + 100 at ruby.c:2026 frame #14: 0x00000001001e084e minirubyprocess_options(argc=0, argv=0x00007fff5fbfdc00, opt=0x00007fff5fbfda28) + 3454 at ruby.c:1682 frame #15: 0x00000001001dfaae minirubyruby_process_options(argc=2, argv=0x00007fff5fbfdbf0) + 238 at ruby.c:2257 frame #16: 0x000000010009ff43 minirubyruby_options(argc=2, argv=0x00007fff5fbfdbf0) + 211 at eval.c:105 frame #17: 0x0000000100000989 minirubymain(argc=2, argv=0x00007fff5fbfdbf0) + 105 at main.c:42 frame #18: 0x00007fff88eda5ad libdyld.dylibstart + 1 (lldb) up 4 frame #4: 0x00000001000bdc61 minirubyrb_raw_obj_info(buff="0x0000000100f46450 [0] T_IMEMO", buff_size=256, obj=4310983760) + 2353 at gc.c:9376 9373 #undef IMEMO_NAME 9374 default: UNREACHABLE; 9375 } -> 9376 snprintf(buff, buff_size, "%s %s", buff, imemo_name); 9377 9378 switch (imemo_type(obj)) { 9379 case imemo_ment: { (lldb) p imemo_name (const char *) $0 = 0xffffffffffffffff (lldb) p imemo_type(obj) (imemo_type) $1 = imemo_ast (lldb)

Revision 61566 - 01/02/2018 06:42 AM - shyouhei (Shyouhei Urabe)

fix SEGV touching uninitialized local variable

This imemo_name is used uninitialized because the switch
above does not cover all possible imemo types.

(lldb) run
Process 26068 launched: './miniruby' (x86_64)
Process 26068 stopped

- thread #1: tid = 0x14ba96, 0x00007fff8a402132 libsystem_c.dylibstrlen + 18, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0xffffffffffffffff0) frame #0: 0x00007fff8a402132 libsystem_c.dylibstrlen + 18 libsystem_c.dylib`strlen: -> 0x7fff8a402132 <+18>: pcmpeqb (%rdi), %xmm0 0x7fff8a402136 <+22>: pmovmskb %xmm0, %esi 0x7fff8a40213a <+26>: andq $0xf, %rcx 0x7fff8a40213e <+30>: orq $-0x1, %rax (lldb) bt
- thread #1: tid = 0x14ba96, 0x00007fff8a402132 libsystem_c.dylib`strlen + 18, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0xffffffffffffffff0)
  - frame #0: 0x00007fff8a402132 libsystem_c.dylibstrlen + 18 frame #1: 0x00000001001f1531 minirubyBSD_vfprintf(fp=0x00007fff5fbfc9e0, fmt0="%s %s", ap=0x00007fff5fbfcbf0) + 5873 at vsnprintf.c:1026 frame #2: 0x00000001001ef213 minirubyruby_do_vsnprintf(str="0x0000000100f46450 [0] T_IMEMO", n=256, fmt="%s %s", ap=0x00007fff5fbfcbf0) + 131 at sprintf.c:1285 frame #3: 0x00000001001ef3ea minirubyruby_snprintf(str="0x0000000100f46450 [0] T_IMEMO", n=256, fmt="%s %s") + 426 at sprintf.c:1300 frame #4: 0x00000001000bdc61 minirubyrb_raw_obj_info(buff="0x0000000100f46450 [0] T_IMEMO", buff_size=256, obj=4310983760) + 2353 at gc.c:9376 frame #5: 0x00000001000b7bff minirubyobj_info(obj=4310983760) + 95 at gc.c:9428 frame #6: 0x00000001000c1658 minirubynewobj_init(klass=0, flags=36890, v1=0, v2=4303040512, v3=4310983800, wb_protected=1, objspace=0x00000001007ee280, obj=4310983760) + 424 at gc.c:1887 frame #7: 0x00000001000b4469 minirubynewobj_of(klass=0, flags=36890, v1=0, v2=4303040512, v3=4310983800, wb_protected=1) + 217 at gc.c:1970 frame #8: 0x00000001000b45eb minirubyrb_imemo_new(type=imemo_ast, v1=0, v2=4303040512, v3=4310983800, v0=0) + 75 at gc.c:2017 frame #9: 0x000000010011daed minirubyrb_ast_new + 61 at node.c:1146 frame #10: 0x0000000100160e15 minirubyrb_parser_compile_file_path(vparser=4310984400, fname=4310984960, file=4310984080, start=1) + 53 at parse.y:5776 frame #11: 0x00000001001e18ea minirubyload_file_internal(argp_v=140734799795024) + 1834 at ruby.c:1907 frame #12: 0x00000001000a1bb5 minirubyrb_ensure(b_proc=(minirubyload_file_internal at ruby.c:1795), data1=140734799795024, e_proc=(minirubyrestore_load_file at ruby.c:2007), data2=140734799795024) + 245 at eval.c:1037 frame #13: 0x00000001001df4a4 minirubyload_file(parser=4310984400, fname=4310984960, f=4310984080, script=1, opt=0x00007fff5fbfda28) + 100 at ruby.c:2026 frame #14: 0x00000001001e084e minirubyprocess_options(argc=0, argv=0x00007fff5fbfdc00, opt=0x00007fff5fbfda28) + 3454 at ruby.c:1682 frame #15: 0x00000001001dfaae minirubyruby_process_options(argc=2, argv=0x00007fff5fbfdbf0) + 238 at ruby.c:2257 frame #16: 0x000000010009ff43 minirubyruby_options(argc=2, argv=0x00007fff5fbfdbf0) + 211 at eval.c:105 frame #17: 0x0000000100000989 minirubymain(argc=2, argv=0x00007fff5fbfdbf0) + 105 at main.c:42 frame #18: 0x00007fff88eda5ad libdyld.dylibstart + 1 (lldb) up 4 frame #4: 0x00000001000bdc61 minirubyrb_raw_obj_info(buff="0x0000000100f46450 [0] T_IMEMO", buff_size=256, obj=4310983760) + 2353 at gc.c:9376 9373 #undef IMEMO_NAME 9374 default: UNREACHABLE; 9375 } -> 9376 snprintf(buff, buff_size, "%s %s", buff, imemo_name); 9377 9378 switch (imemo_type(obj)) { 9379 case imemo_ment: { (lldb) p imemo_name (const char *) $0 = 0xffffffffffffffff (lldb) p imemo_type(obj) (imemo_type) $1 = imemo_ast (lldb)

Revision 61568 - 01/02/2018 06:42 AM - shyouhei (Shyouhei Urabe)

fix SEGV inspecting already freed objects

obj_info() assumes the given object is alive. Passing freed
objects to it results in SEGV.

(lldb) run
Process 29718 launched: './miniruby' (x86_64)
Process 29718 stopped

- thread #1: tid = 0x3082c5, 0x00000001000bfaab minirubypathobj_path(pathobj=4478683640) + 70 at vm_core.h:269, queue =

'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0) frame #0: 0x00000001000bfaab minirubypathobj_path(pathobj=4478683640) + 70 at vm_core.h:269 266 } 267 else { 268 VM_ASSERT(RB_TYPE_P(pathobj, T_ARRAY)); -> 269 return RARRAY_AREF(pathobj, PATHOBJ_PATH); 270 } 271 } 272 (lldb) bt

- thread #1: tid = 0x3082c5, 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:269, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
  - frame #0: 0x00000001000bfaab minirubypathobj_path(pathobj=4478683640) + 70 at vm_core.h:269 frame #1: 0x00000001000c25ff minirubyrb_iseq_path(iseq=0x000000010af34a20) + 32 at iseq.c:723 frame #2: 0x000000010009db09 minirubyrb_raw_iseq_info(buff="0x000000010af34a20 [1] T_IMEMO iseq", buff_size=256, iseq=0x000000010af34a20) + 69 at gc.c:9274 frame #3: 0x000000010009e45a minirubyrb_raw_obj_info(buff="0x000000010af34a20 [1] T_IMEMO iseq", buff_size=256, obj=4478683680) + 2191 at gc.c:9397 frame #4: 0x000000010009e4d5 minirubyobj_info(obj=4478683680) + 98 at gc.c:9429 frame #5: 0x0000000100091ae3 minirubygc_page_sweep(objspace=0x00000001007d3280, heap=0x00000001007d32a0, sweep_page=0x000000010ae07bc0) + 622 at gc.c:3529 frame #6: 0x000000010009206a minirubygc_sweep_step(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 188 at gc.c:3705 frame #7: 0x0000000100092254 minirubygc_sweep_continue(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 133 at gc.c:3772 frame #8: 0x000000010008d7f9 minirubyheap_prepare(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 48 at gc.c:1746 frame #9: 0x000000010008d8a1 minirubyheap_get_freeobj_from_next_freepage(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 37 at gc.c:1769 frame #10: 0x000000010008d98d minirubyheap_get_freeobj(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 83 at gc.c:1803 frame #11: 0x000000010008dcb0 minirubynewobj_slowpath(klass=4334386280, flags=5, v1=0, v2=0, v3=0, objspace=0x00000001007d3280, wb_protected=1) + 220 at gc.c:1930 frame #12: 0x000000010008dd6c minirubynewobj_slowpath_wb_protected(klass=4334386280, flags=5, v1=0, v2=0, v3=0, objspace=0x00000001007d3280) + 76 at gc.c:1942 frame #13: 0x000000010008dea1 minirubynewobj_of(klass=4334386280, flags=5, v1=0, v2=0, v3=0, wb_protected=1) + 221 at gc.c:1974 frame #14: 0x000000010008df39 minirubyrb_wb_protected_newobj_of(klass=4334386280, flags=5) + 54 at gc.c:1990 frame #15: 0x0000000100195f7c minirubystr_alloc(klass=4334386280) + 29 at string.c:692 frame #16: 0x0000000100195fe9 minirubystr_new0(klass=4334386280, ptr="gitm", len=4, termlen=1) + 73 at string.c:714 frame #17: 0x000000010019633e minirubyrb_enc_str_new(ptr="gitm", len=4, enc=0x00000001025d50a0) + 81 at string.c:766 frame #18: 0x000000010010a80a minirubyparser_str_new(p="gitm", n=4, enc=0x00000001025d50a0, func=66, enc0=0x00000001025d50a0) + 50 at parse.y:5817 frame #19: 0x000000010010ce1a minirubyparser_parse_string(parser=0x00000001042ac5c0, quote=0x000000010460c028) + 795 at parse.y:6675 frame #20: 0x0000000100120bd minirubyparser_yylex(parser=0x00000001042ac5c0) + 159 at parse.y:8281 frame #21: 0x0000000100115068 minirubyyylex(lval=0x00007fff5fbf9948, yylloc=0x00007fff5fbf9ab0, parser=0x00000001042ac5c0) + 55 at parse.y:8931 frame #22: 0x00000001000fc79f minirubyruby_yyparse(parser=0x00000001042ac5c0) + 1198 at parse.c:5798 frame #23: 0x0000000100109f5a minirubyyycompile0(arg=4364879296) + 317 at parse.y:5595 frame #24: 0x0000000100214ef0 minirubyrb_suppress_tracing(func=(minirubyyycompile0 at parse.y:5565), arg=4364879296) + 349 at vm_trace.c:397 frame #25: 0x000000010010a1df minirubyyycompile(parser=0x00000001042ac5c0, fname=4443743440, line=1) + 126 at parse.y:5637 frame #26: 0x000000010010a4c1 minirubyparser_compile_string(vparser=4443743480, fname=4443743440, s=4443743520, line=1) + 191 at parse.y:5706 frame #27: 0x000000010010a5b7 minirubyrb_parser_compile_string_path(vparser=4443743480, f=4443743440, s=4443743520, line=1) + 58 at parse.y:5730 frame #28: 0x0000000100206025 minirubyeval_make_iseq(src=4443743520, fname=4443743440, line=1, bind=0x0000000000000000, base_block=0x00007fff5fbfb370) + 266 at vm_eval.c:1274 frame #29: 0x0000000100206153 minirubyeval_string_with_cref(self=4334412520, src=4443743520, cref=0x0000000000000000, file=52, line=1) + 197 at vm_eval.c:1307 frame #30: 0x0000000100206389 minirubyrb_f_eval(argc=1, argv=0x0000000102400eb8, self=4334412520) + 219 at vm_eval.c:1382 frame #31: 0x00000001001f247c minirubycall_cfunc_m1(func=(minirubyrb_f_eval at vm_eval.c:1364), recv=4334412520, argc=1, argv=0x0000000102400eb8) + 47 at vm_insnhelper.c:1723 frame #32: 0x00000001001f2f87 minirubyvm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x0000000102500d80, calling=0x00007fff5fbfbf50, ci=0x000000010263f240, cc=0x0000000100749b50) + 386 at vm_insnhelper.c:1918 frame #33: 0x00000001001f30d6 minirubyvm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500d80, calling=0x00007fff5fbfbf50, ci=0x000000010263f240, cc=0x0000000100749b50) + 149 at vm_insnhelper.c:1934 frame #34: 0x00000001001faf0e minirubyvm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:915 frame #35: 0x000000010020b75d minirubyvm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771 frame #36: 0x0000000102093f8 minirubyinvoke_block(ec=0x00000001007d3548, iseq=0x000000010252d7f0, self=4334412520, captured=0x0000000102500df8, cref=0x0000000000000000, type=572653569, opt_pc=0) + 224 at vm.c:988 frame #37: 0x0000000100209766 minirubyinvoke_iseq_block_from_c(ec=0x00000001007d3548, captured=0x0000000102500df8, self=4334412520, argc=0, argv=0x0000000000000000, passed_block_handler=0, cref=0x0000000000000000, is_lambda=0) + 389 at vm.c:1040 frame #38: 0x0000000100209824 minirubyinvoke_block_from_c_bh(ec=0x00000001007d3548, block_handler=4333768185, argc=0, argv=0x0000000000000000, passed_block_handler=0, cref=0x0000000000000000, is_lambda=0, force_blockarg=0) + 138 at vm.c:1058 frame #39: 0x00000001002099d0 minirubyvm_yield(ec=0x00000001007d3548, argc=0, argv=0x0000000000000000) + 69 at vm.c:1103 frame #40: 0x0000000100205623 minirubyrb_yield_0(argc=0, argv=0x0000000000000000) + 40 at vm_eval.c:970 frame #41: 0x0000000100205964 minirubyloop_i + 19 at vm_eval.c:1049 frame #42: 0x000000010007db07 minirubyrb_rescue2(b_proc=(minirubyloop_i at vm_eval.c:1047), data1=0, r_proc=(minirubyloop_stop at vm_eval.c:1056), data2=0) + 369 at eval.c:896 frame #43: 0x0000000100205a2e minirubyrb_f_loop(self=4334412520) + 121 at vm_eval.c:1100 frame #44: 0x00000001001f24a7 minirubycall_cfunc_0(func=(minirubyrb_f_loop at vm_eval.c:1098), recv=4334412520, argc=0, argv=0x0000000102400e80) + 41 at vm_insnhelper.c:1729 frame #45: 0x00000001001f2f87 minirubyvm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x0000000102500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 386 at vm_insnhelper.c:1918 frame #46: 0x00000001001f30d6 minirubyvm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 149 at vm_insnhelper.c:1934 frame #47: 0x00000001001f4319 minirubyvm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 239 at vm_insnhelper.c:2232 frame #48: 0x00000001001f4a2c minirubyvm_call_method(ec=0x00000001007d3548, cfp=0x0000000102500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 253 at vm_insnhelper.c:2366 frame #49: 0x00000001001f4b7a minirubyvm_call_general(ec=0x00000001007d3548, reg_cfp=0x0000000102500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 59 at vm_insnhelper.c:2398 frame #50: 0x00000001001fab2f minirubyvm_exec_core(ec=0x00000001007d3548, initial=0) + 7480 at insns.def:850 frame #51: 0x000000010020b75d minirubyvm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771 frame #52: 0x000000010020c40f minirubyrb_iseq_eval_main(iseq=0x000000010252dd90) + 52 at vm.c:2019 frame #53: 0x000000010007c768 minirubyruby_exec_internal(n=0x000000010252dd90) + 297 at eval.c:246 frame #54: 0x000000010007c88e

minirubyruby_exec_node(n=0x000000010252dd90) + 36 at eval.c:310 frame #55: 0x000000010007c861 minirubyruby_run_node(n=0x000000010252dd90) + 62 at eval.c:302 frame #56: 0x000000010000138d minirubymain(argc=2, argv=0x00007fff5fbfdbf0) + 113 at main.c:42 frame #57: 0x00007fff88eda5ad libdyld.dylib`start + 1 (lldb) p ((struct RVALUE*)pathobj)->as.basic (RBasic) $0 = (flags = 0, klass = 4478683600) (lldb)

Revision 61568 - 01/02/2018 06:42 AM - shyouhei (Shyouhei Urabe)

fix SEGV inspecting already freed objects

obj_info() assumes the given object is alive. Passing freed
objects to it results in SEGV.

(lldb) run
Process 29718 launched: './miniruby' (x86_64)
Process 29718 stopped

- thread #1: tid = 0x3082c5, 0x00000001000bfaab minirubypathobj_path(pathobj=4478683640) + 70 at vm_core.h:269, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0) frame #0: 0x00000001000bfaab minirubypathobj_path(pathobj=4478683640) + 70 at vm_core.h:269 266 } 267 else { 268 VM_ASSERT(RB_TYPE_P(pathobj, T_ARRAY)); -> 269 return RARRAY_AREF(pathobj, PATHOBJ_PATH); 270 } 271 } 272 (lldb) bt
- thread #1: tid = 0x3082c5, 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:269, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
  - frame #0: 0x00000001000bfaab minirubypathobj_path(pathobj=4478683640) + 70 at vm_core.h:269 frame #1: 0x00000001000c25ff minirubyrb_iseq_path(iseq=0x000000010af34a20) + 32 at iseq.c:723 frame #2: 0x0000000100009db09 minirubyrb_raw_iseq_info(buff="0x000000010af34a20 [1] T_IMEMO iseq", buff_size=256, iseq=0x000000010af34a20) + 69 at gc.c:9274 frame #3: 0x000000010009e45a minirubyrb_raw_obj_info(buff="0x000000010af34a20 [1] T_IMEMO iseq", buff_size=256, obj=4478683680) + 2191 at gc.c:9397 frame #4: 0x000000010009e4d5 minirubyobj_info(obj=4478683680) + 98 at gc.c:9429 frame #5: 0x0000000100091ae3 minirubygc_page_sweep(objspace=0x00000001007d3280, heap=0x00000001007d32a0, sweep_page=0x000000010ae07bc0) + 622 at gc.c:3529 frame #6: 0x000000010009206a minirubygc_sweep_step(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 188 at gc.c:3705 frame #7: 0x0000000100092254 minirubygc_sweep_continue(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 133 at gc.c:3772 frame #8: 0x000000010008d7f9 minirubyheap_prepare(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 48 at gc.c:1746 frame #9: 0x000000010008d8a1 minirubyheap_get_freeobj_from_next_freepage(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 37 at gc.c:1769 frame #10: 0x000000010008d98d minirubyheap_get_freeobj(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 83 at gc.c:1803 frame #11: 0x000000010008dcb0 minirubynewobj_slowpath(klass=4334386280, flags=5, v1=0, v2=0, v3=0, objspace=0x00000001007d3280, wb_protected=1) + 220 at gc.c:1930 frame #12: 0x000000010008dd6c minirubynewobj_slowpath_wb_protected(klass=4334386280, flags=5, v1=0, v2=0, v3=0, objspace=0x00000001007d3280) + 76 at gc.c:1942 frame #13: 0x000000010008dea1 minirubynewobj_of(klass=4334386280, flags=5, v1=0, v2=0, v3=0, wb_protected=1) + 221 at gc.c:1974 frame #14: 0x000000010008df39 minirubyrb_wb_protected_newobj_of(klass=4334386280, flags=5) + 54 at gc.c:1990 frame #15: 0x0000000100195f7c minirubystr_alloc(klass=4334386280) + 29 at string.c:692 frame #16: 0x0000000100195fe9 minirubystr_new0(klass=4334386280, ptr="gitm", len=4, termlen=1) + 73 at string.c:714 frame #17: 0x000000010019633e minirubyrb_enc_str_new(ptr="gitm", len=4, enc=0x00000001025d50a0) + 81 at string.c:766 frame #18: 0x000000010010a80a minirubyparser_str_new(p="gitm", n=4, enc=0x00000001025d50a0, func=66, enc0=0x00000001025d50a0) + 50 at parse.y:5817 frame #19: 0x000000010010ce1a minirubyparser_parse_string(parser=0x00000001042ac5c0, quote=0x000000010460c028) + 795 at parse.y:6675 frame #20: 0x0000000100120bd minirubyparser_yylex(parser=0x00000001042ac5c0) + 159 at parse.y:8281 frame #21: 0x0000000100115068 minirubyyylex(lval=0x00007fff5fbf9948, yylloc=0x00007fff5fbf9ab0, parser=0x00000001042ac5c0) + 55 at parse.y:8931 frame #22: 0x00000001000fc79f minirubyruby_yyparse(parser=0x00000001042ac5c0) + 1198 at parse.c:5798 frame #23: 0x0000000100109f5a minirubyyycompile0(arg=4364879296) + 317 at parse.y:5595 frame #24: 0x0000000100214ef0 minirubyrb_suppress_tracing(func=(minirubyyycompile0 at parse.y:5565), arg=4364879296) + 349 at vm_trace.c:397 frame #25: 0x000000010010a1df minirubyyycompile(parser=0x00000001042ac5c0, fname=4443743440, line=1) + 126 at parse.y:5637 frame #26: 0x000000010010a4c1 minirubyparser_compile_string(vparser=4443743480, fname=4443743440, s=4443743520, line=1) + 191 at parse.y:5706 frame #27: 0x000000010010a5b7 minirubyrb_parser_compile_string_path(vparser=4443743480, f=4443743440, s=4443743520, line=1) + 58 at parse.y:5730 frame #28: 0x0000000100206025 minirubyeval_make_iseq(src=4443743520, fname=4443743440, line=1, bind=0x0000000000000000, base_block=0x00007fff5fbfb370) + 266 at vm_eval.c:1274 frame #29: 0x0000000100206153 minirubyeval_string_with_cref(self=4334412520, src=4443743520, cref=0x0000000000000000, file=52, line=1) + 197 at vm_eval.c:1307 frame #30: 0x0000000100206389 minirubyrb_f_eval(argc=1, argv=0x0000000102400eb8, self=4334412520) + 219 at vm_eval.c:1382 frame #31: 0x00000001001f247c minirubycall_cfunc_m1(func=(minirubyrb_f_eval at vm_eval.c:1364), recv=4334412520, argc=1, argv=0x0000000102400eb8) + 47 at vm_insnhelper.c:1723 frame #32: 0x00000001001f2f87 minirubyvm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x0000000102500d80, calling=0x00007fff5fbfbf50, ci=0x000000010263f240, cc=0x0000000100749b50) + 386 at vm_insnhelper.c:1918 frame #33: 0x00000001001f30d6 minirubyvm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500d80, calling=0x00007fff5fbfbf50, ci=0x000000010263f240, cc=0x0000000100749b50) + 149 at vm_insnhelper.c:1934 frame #34: 0x00000001001faf0e minirubyvm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:915 frame #35: 0x000000010020b75d minirubyvm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771 frame #36: 0x00000001002093f8 minirubyinvoke_block(ec=0x00000001007d3548, iseq=0x000000010252d7f0, self=4334412520, captured=0x0000000102500df8, cref=0x0000000000000000, type=572653569, opt_pc=0) + 224 at vm.c:988 frame #37: 0x0000000100209766 minirubyinvoke_iseq_block_from_c(ec=0x00000001007d3548, captured=0x0000000102500df8, self=4334412520, argc=0, argv=0x0000000000000000, passed_block_handler=0, cref=0x0000000000000000, is_lambda=0) + 389 at vm.c:1040 frame #38: 0x0000000100209824 minirubyinvoke_block_from_c_bh(ec=0x00000001007d3548, block_handler=4333768185, argc=0, argv=0x0000000000000000, passed_block_handler=0, cref=0x0000000000000000, is_lambda=0, force_blockarg=0) + 138 at vm.c:1058 frame #39: 0x00000001002099d0 minirubyvm_yield(ec=0x00000001007d3548, argc=0, argv=0x0000000000000000) + 69 at vm.c:1103 frame #40: 0x0000000100205623 minirubyrb_yield_0(argc=0, argv=0x0000000000000000) + 40 at vm_eval.c:970 frame #41: 0x0000000100205964 minirubyloop_i + 19 at vm_eval.c:1049 frame #42: 0x000000010007db07 minirubyrb_rescue2(b_proc=(minirubyloop_i at vm_eval.c:1047), data1=0, r_proc=(minirubyloop_stop at vm_eval.c:1056), data2=0) + 369

at eval.c:896 frame #43: 0x0000000100205a2e minirubyrb_f_loop(self=4334412520) + 121 at vm_eval.c:1100 frame #44: 0x00000001001f24a7 minirubycall_cfunc_0(func=(minirubyrb_f_loop at vm_eval.c:1098), recv=4334412520, argc=0, argv=0x0000000102400e80) + 41 at vm_insnhelper.c:1729 frame #45: 0x00000001001f2f87 minirubyvm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x0000000102500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 386 at vm_insnhelper.c:1918 frame #46: 0x00000001001f30d6 minirubyvm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 149 at vm_insnhelper.c:1934 frame #47: 0x00000001001f4319 minirubyvm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 239 at vm_insnhelper.c:2232 frame #48: 0x00000001001f4a2c minirubyvm_call_method(ec=0x00000001007d3548, cfp=0x0000000102500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 253 at vm_insnhelper.c:2366 frame #49: 0x00000001001f4b7a minirubyvm_call_general(ec=0x00000001007d3548, reg_cfp=0x0000000102500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 59 at vm_insnhelper.c:2398 frame #50: 0x00000001001fab2f minirubyvm_exec_core(ec=0x00000001007d3548, initial=0) + 7480 at insns.def:850 frame #51: 0x000000010020b75d minirubyvm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771 frame #52: 0x000000010020c40f minirubyrb_iseq_eval_main(iseq=0x000000010252dd90) + 52 at vm.c:2019 frame #53: 0x0000000010007c768 minirubyruby_exec_internal(n=0x000000010252dd90) + 297 at eval.c:246 frame #54: 0x0000000010007c88e minirubyruby_exec_node(n=0x000000010252dd90) + 36 at eval.c:310 frame #55: 0x0000000010007c861 minirubyruby_run_node(n=0x000000010252dd90) + 62 at eval.c:302 frame #56: 0x000000010000138d minirubymain(argc=2, argv=0x00007fff5fbfdbf0) + 113 at main.c:42 frame #57: 0x00007fff88eda5ad libdyld.dylib`start + 1 (lldb) p ((struct RVALUE*)pathobj)->as.basic (RBasic) $0 = (flags = 0, klass = 4478683600) (lldb)

Revision 61569 - 01/02/2018 06:42 AM - shyouhei (Shyouhei Urabe)

fix SEGV inspecting uninitialized objects

obj_info() assumes the given object is alive. OTOH
gc_writebarrier_incremental is called before or in middle of
object initialization. Can casue SEGV.

(lldb) run
Process 48188 launched: './miniruby' (x86_64)
Process 48188 stopped

- thread #1: tid = 0x30fd53, 0x00000001000bf7a9 minirubyrb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=EXC_I386_GPFLT) frame #0: 0x00000001000bf7a9 minirubyrb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072 2069 static inline const VALUE * 2070 rb_array_const_ptr(VALUE a) 2071 { -> 2072 return FIX_CONST_VALUE_PTR((RBASIC(a)->flags & RARRAY_EMBED_FLAG) ? 2073 RARRAY(a)->as.ary : RARRAY(a)->as.heap.ptr); 2074 } 2075 (lldb) bt
- thread #1: tid = 0x30fd53, 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=EXC_I386_GPFLT)
  - frame #0: 0x00000001000bf7a9 minirubyrb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072 frame #1: 0x00000001000bfaab minirubypathobj_path(pathobj=5251291222225483145) + 70 at vm_core.h:269 frame #2: 0x00000001000c25ff minirubyrb_iseq_path(iseq=0x00000001025b71a8) + 32 at iseq.c:723 frame #3: 0x000000010009db09 minirubyrb_raw_iseq_info(buff="0x00000001025b7158 [0] proc (Proc)", buff_size=256, iseq=0x00000001025b71a8) + 69 at gc.c:9274 frame #4: 0x000000010009e1d5 minirubyrb_raw_obj_info(buff="0x00000001025b7158 [0] proc (Proc)", buff_size=256, obj=4334514520) + 1546 at gc.c:9351 frame #5: 0x000000010009e4d5 minirubyobj_info(obj=4334514520) + 98 at gc.c:9429 frame #6: 0x0000000100096658 minirubygc_writebarrier_incremental(a=4334514520, b=4334514600, objspace=0x00000001007d3280) + 61 at gc.c:5963 frame #7: 0x00000001000968ca minirubyrb_gc_writebarrier(a=4334514520, b=4334514600) + 127 at gc.c:6009 frame #8: 0x00000001001eabe0 minirubyrb_obj_written(a=4334514520, oldv=52, b=4334514600, filename="/Users/urabe.shyouhei/data/src/pedantic/vm.c", line=821) + 72 at ruby.h:1472 frame #9: 0x00000001001eac2c minirubyrb_obj_write(a=4334514520, slot=0x000000010259ff10, b=4334514600, filename="/Users/urabe.shyouhei/data/src/pedantic/vm.c", line=821) + 70 at ruby.h:1489 frame #10: 0x0000000100208b6f minirubyvm_proc_create_from_captured(klass=4311027960, captured=0x0000000102500338, block_type=block_type_ifunc, is_from_method='\0', is_lambda='\x01') + 137 at vm.c:821 frame #11: 0x0000000100208e5c minirubyrb_vm_make_proc_lambda(ec=0x00000001007d3548, captured=0x0000000102500338, klass=4311027960, is_lambda='\x01') + 134 at vm.c:892 frame #12: 0x000000010011f08e minirubyproc_new(klass=4311027960, is_lambda='\x01') + 445 at proc.c:752 frame #13: 0x000000010011f110 minirubyrb_block_lambda + 27 at proc.c:808 frame #14: 0x00000001001f24a7 minirubycall_cfunc_0(func=(minirubyrb_block_lambda at proc.c:807), recv=4310991600, argc=0, argv=0x0000000000000000) + 41 at vm_insnhelper.c:1729 frame #15: 0x00000001002033de minirubyvm_call0_cfunc_with_frame(ec=0x00000001007d3548, calling=0x00007fff5fbfb080, ci=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 370 at vm_eval.c:85 frame #16: 0x00000001002034d9 minirubyvm_call0_cfunc(ec=0x00000001007d3548, calling=0x00007fff5fbfb080, ci=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 59 at vm_eval.c:100 frame #17: 0x000000010020368f minirubyvm_call0_body(ec=0x00000001007d3548, calling=0x00007fff5fbfb080, ci=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 436 at vm_eval.c:131 frame #18: 0x000000010020326a minirubyvm_call0(ec=0x00000001007d3548, recv=4310991600, id=2993, argc=0, argv=0x0000000000000000, me=0x0000000100f48110) + 142 at vm_eval.c:58 frame #19: 0x0000000100203c60 minirubyrb_call0(ec=0x00000001007d3548, recv=4310991600, mid=2993, argc=0, argv=0x0000000000000000, scope=CALL_FCALL, self=4334514640) + 166 at vm_eval.c:296 frame #20: 0x0000000100204827 minirubyrb_call(recv=4310991600, mid=2993, argc=0, argv=0x0000000000000000, scope=CALL_FCALL) + 84 at vm_eval.c:589 frame #21: 0x000000010020518b minirubyrb_funcallv(recv=4310991600, mid=2993, argc=0, argv=0x0000000000000000) + 52 at vm_eval.c:815 frame #22: 0x000000010012242e minirubymlambda(method=0) + 45 at proc.c:2661 frame #23: 0x0000000100205bac minirubyrb_iterate0(it_proc=(minirubymlambda at proc.c:2660), data1=0, ifunc=0x00000001025b71a8, ec=0x00000001007d3548) + 380 at vm_eval.c:1134 frame #24: 0x0000000100205d16 minirubyrb_iterate(it_proc=(minirubymlambda at proc.c:2660), data1=0, bl_proc=(minirubybmcall at proc.c:2666), data2=4334514640) + 88 at vm_eval.c:1166 frame #25: 0x00000001001224c7 minirubymethod_to_proc(method=4334514640) + 43 at proc.c:2701 frame #26: 0x00000001001f24a7 minirubycall_cfunc_0(func=(minirubymethod_to_proc at proc.c:2688), recv=4334514640, argc=0, argv=0x0000000102400568) + 41 at

vm_insnhelper.c:1729 frame #27: 0x00000001001f2f87 minirubyvm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x0000000102500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 386 at vm_insnhelper.c:1918 frame #28: 0x00000001001f30d6 minirubyvm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 149 at vm_insnhelper.c:1934 frame #29: 0x00000001001f4319 minirubyvm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 239 at vm_insnhelper.c:2232 frame #30: 0x00000001001f49a4 minirubyvm_call_method(ec=0x00000001007d3548, cfp=0x0000000102500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 117 at vm_insnhelper.c:2355 frame #31: 0x00000001001f4b7a minirubyvm_call_general(ec=0x00000001007d3548, reg_cfp=0x0000000102500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 59 at vm_insnhelper.c:2398 frame #32: 0x00000001001faf0e minirubyvm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:915 frame #33: 0x000000010020b75d minirubyvm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771 frame #34: 0x000000010020c3d1 minirubyrb_iseq_eval(iseq=0x00000001007f8270) + 52 at vm.c:2008 frame #35: 0x00000001000caa4a minirubyrb_load_internal0(ec=0x00000001007d3548, fname=4310799960, wrap=0) + 631 at load.c:611 frame #36: 0x00000001000cab36 minirubyrb_load_internal(fname=4310799960, wrap=0) + 46 at load.c:642 frame #37: 0x00000001000cae1d minirubyrb_f_load(argc=1, argv=0x00000001024004b8) + 217 at load.c:710 frame #38: 0x00000001001f247c minirubycall_cfunc_m1(func=(minirubyrb_f_load at load.c:695), recv=4311327440, argc=1, argv=0x00000001024004b8) + 47 at vm_insnhelper.c:1723 frame #39: 0x00000001001f2f87 minirubyvm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x00000001025003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 386 at vm_insnhelper.c:1918 frame #40: 0x00000001001f30d6 minirubyvm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x00000001025003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 149 at vm_insnhelper.c:1934 frame #41: 0x00000001001f4319 minirubyvm_call_method_each_type(ec=0x00000001007d3548, cfp=0x00000001025003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 239 at vm_insnhelper.c:2232 frame #42: 0x00000001001f4a2c minirubyvm_call_method(ec=0x00000001007d3548, cfp=0x00000001025003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 253 at vm_insnhelper.c:2366 frame #43: 0x00000001001f4b7a minirubyvm_call_general(ec=0x00000001007d3548, reg_cfp=0x00000001025003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 59 at vm_insnhelper.c:2398 frame #44: 0x00000001001faf0e minirubyvm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:915 frame #45: 0x000000010020b75d minirubyvm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771 frame #46: 0x000000010020c40f minirubyrb_iseq_eval_main(iseq=0x0000000100f21240) + 52 at vm.c:2019 frame #47: 0x000000010007c774 minirubyruby_exec_internal(n=0x0000000100f21240) + 297 at eval.c:246 frame #48: 0x000000010007c89a minirubyruby_exec_node(n=0x0000000100f21240) + 36 at eval.c:310 frame #49: 0x000000010007c86d minirubyruby_run_node(n=0x0000000100f21240) + 62 at eval.c:302 frame #50: 0x0000000100001399 minirubymain(argc=9, argv=0x00007fff5fbfdae0) + 113 at main.c:42 frame #51: 0x00007fff88eda5ad libdyld.dylibstart + 1 (lldb)

Revision 61569 - 01/02/2018 06:42 AM - shyouhei (Shyouhei Urabe)

fix SEGV inspecting uninitialized objects

obj_info() assumes the given object is alive. OTOH
gc_writebarrier_incremental is called before or in middle of
object initialization. Can casue SEGV.

(lldb) run
Process 48188 launched: './miniruby' (x86_64)
Process 48188 stopped

- thread #1: tid = 0x30fd53, 0x00000001000bf7a9 minirubyrb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=EXC_I386_GPFLT) frame #0: 0x00000001000bf7a9 minirubyrb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072 2069 static inline const VALUE * 2070 rb_array_const_ptr(VALUE a) 2071 { -> 2072 return FIX_CONST_VALUE_PTR((RBASIC(a)->flags & RARRAY_EMBED_FLAG) ? 2073 RARRAY(a)->as.ary : RARRAY(a)->as.heap.ptr); 2074 } 2075 (lldb) bt
- thread #1: tid = 0x30fd53, 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=EXC_I386_GPFLT)
  - frame #0: 0x00000001000bf7a9 minirubyrb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072 frame #1: 0x00000001000bfaab minirubypathobj_path(pathobj=5251291222225483145) + 70 at vm_core.h:269 frame #2: 0x00000001000c25ff minirubyrb_iseq_path(iseq=0x00000001025b71a8) + 32 at iseq.c:723 frame #3: 0x000000010009db09 minirubyrb_raw_iseq_info(buff="0x00000001025b7158 [0] proc (Proc)", buff_size=256, iseq=0x00000001025b71a8) + 69 at gc.c:9274 frame #4: 0x000000010009e1d5 minirubyrb_raw_obj_info(buff="0x00000001025b7158 [0] proc (Proc)", buff_size=256, obj=4334514520) + 1546 at gc.c:9351 frame #5: 0x000000010009e4d5 minirubyobj_info(obj=4334514520) + 98 at gc.c:9429 frame #6: 0x0000000100096658 minirubygc_writebarrier_incremental(a=4334514520, b=4334514600, objspace=0x00000001007d3280) + 61 at gc.c:5963 frame #7: 0x00000001000968ca minirubyrb_gc_writebarrier(a=4334514520, b=4334514600) + 127 at gc.c:6009 frame #8: 0x00000001001eabe0 minirubyrb_obj_written(a=4334514520, oldv=52, b=4334514600, filename="/Users/urabe.shyouhei/data/src/pedantic/vm.c", line=821) + 72 at ruby.h:1472 frame #9: 0x00000001001eac2c minirubyrb_obj_write(a=4334514520, slot=0x000000010259ff10, b=4334514600, filename="/Users/urabe.shyouhei/data/src/pedantic/vm.c", line=821) + 70 at ruby.h:1489 frame #10: 0x0000000100208b6f minirubyvm_proc_create_from_captured(klass=4311027960, captured=0x0000000102500338, block_type=block_type_ifunc, is_from_method='\0', is_lambda='\x01') + 137 at vm.c:821 frame #11: 0x0000000100208e5c minirubyrb_vm_make_proc_lambda(ec=0x00000001007d3548, captured=0x0000000102500338, klass=4311027960, is_lambda='\x01') + 134 at vm.c:892 frame #12: 0x000000010011f08e minirubyproc_new(klass=4311027960, is_lambda='\x01') + 445 at proc.c:752 frame #13: 0x000000010011f110 minirubyrb_block_lambda + 27 at proc.c:808 frame #14: 0x00000001001f24a7 minirubycall_cfunc_0(func=(minirubyrb_block_lambda at proc.c:807), recv=4310991600, argc=0, argv=0x0000000000000000) + 41 at vm_insnhelper.c:1729 frame #15: 0x00000001002033de minirubyvm_call0_cfunc_with_frame(ec=0x00000001007d3548, calling=0x00007fff5fbfb080, ci=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 370 at vm_eval.c:85 frame #16: 0x00000001002034d9 minirubyvm_call0_cfunc(ec=0x00000001007d3548, calling=0x00007fff5fbfb080, ci=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0,

argv=0x0000000000000000) + 59 at vm_eval.c:100 frame #17: 0x000000010020368f minirubyvm_call0_body(ec=0x00000001007d3548, calling=0x00007fff5fbfb080, ci=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 436 at vm_eval.c:131 frame #18: 0x000000010020326a minirubyvm_call0(ec=0x00000001007d3548, recv=4310991600, id=2993, argc=0, argv=0x0000000000000000, me=0x0000000100f48110) + 142 at vm_eval.c:58 frame #19: 0x0000000100203c60 minirubyrb_call0(ec=0x00000001007d3548, recv=4310991600, mid=2993, argc=0, argv=0x0000000000000000, scope=CALL_FCALL, self=4334514640) + 166 at vm_eval.c:296 frame #20: 0x0000000100204827 minirubyrb_call(recv=4310991600, mid=2993, argc=0, argv=0x0000000000000000, scope=CALL_FCALL) + 84 at vm_eval.c:589 frame #21: 0x000000010020518b minirubyrb_funcallv(recv=4310991600, mid=2993, argc=0, argv=0x0000000000000000) + 52 at vm_eval.c:815 frame #22: 0x000000010012242e minirubymlambda(method=0) + 45 at proc.c:2661 frame #23: 0x0000000100205bac minirubyrb_iterate0(it_proc=(minirubymlambda at proc.c:2660), data1=0, ifunc=0x00000001025b71a8, ec=0x00000001007d3548) + 380 at vm_eval.c:1134 frame #24: 0x0000000100205d16 minirubyrb_iterate(it_proc=(minirubymlambda at proc.c:2660), data1=0, bl_proc=(minirubybmcall at proc.c:2666), data2=4334514640) + 88 at vm_eval.c:1166 frame #25: 0x00000001001224c7 minirubymethod_to_proc(method=4334514640) + 43 at proc.c:2701 frame #26: 0x00000001001f24a7 minirubycall_cfunc_0(func=(minirubymethod_to_proc at proc.c:2688), recv=4334514640, argc=0, argv=0x0000000102400568) + 41 at vm_insnhelper.c:1729 frame #27: 0x00000001001f2f87 minirubyvm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x0000000102500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 386 at vm_insnhelper.c:1918 frame #28: 0x00000001001f30d6 minirubyvm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 149 at vm_insnhelper.c:1934 frame #29: 0x00000001001f4319 minirubyvm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 239 at vm_insnhelper.c:2232 frame #30: 0x00000001001f49a4 minirubyvm_call_method(ec=0x00000001007d3548, cfp=0x0000000102500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 117 at vm_insnhelper.c:2355 frame #31: 0x00000001001f4b7a minirubyvm_call_general(ec=0x00000001007d3548, reg_cfp=0x0000000102500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 59 at vm_insnhelper.c:2398 frame #32: 0x00000001001faf0e minirubyvm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:915 frame #33: 0x000000010020b75d minirubyvm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771 frame #34: 0x000000010020c3d1 minirubyrb_iseq_eval(iseq=0x00000001007f8270) + 52 at vm.c:2008 frame #35: 0x00000001000caa4a minirubyrb_load_internal0(ec=0x00000001007d3548, fname=4310799960, wrap=0) + 631 at load.c:611 frame #36: 0x00000001000cab36 minirubyrb_load_internal(fname=4310799960, wrap=0) + 46 at load.c:642 frame #37: 0x00000001000cae1d minirubyrb_f_load(argc=1, argv=0x00000001024004b8) + 217 at load.c:710 frame #38: 0x00000001001f247c minirubycall_cfunc_m1(func=(minirubyrb_f_load at load.c:695), recv=4311327440, argc=1, argv=0x00000001024004b8) + 47 at vm_insnhelper.c:1723 frame #39: 0x00000001001f2f87 minirubyvm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x00000001025003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 386 at vm_insnhelper.c:1918 frame #40: 0x00000001001f30d6 minirubyvm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x00000001025003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 149 at vm_insnhelper.c:1934 frame #41: 0x00000001001f4319 minirubyvm_call_method_each_type(ec=0x00000001007d3548, cfp=0x00000001025003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 239 at vm_insnhelper.c:2232 frame #42: 0x00000001001f4a2c minirubyvm_call_method(ec=0x00000001007d3548, cfp=0x00000001025003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 253 at vm_insnhelper.c:2366 frame #43: 0x00000001001f4b7a minirubyvm_call_general(ec=0x00000001007d3548, reg_cfp=0x00000001025003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 59 at vm_insnhelper.c:2398 frame #44: 0x00000001001faf0e minirubyvm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:915 frame #45: 0x000000010020b75d minirubyvm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771 frame #46: 0x000000010020c40f minirubyrb_iseq_eval_main(iseq=0x0000000100f21240) + 52 at vm.c:2019 frame #47: 0x000000010007c774 minirubyruby_exec_internal(n=0x0000000100f21240) + 297 at eval.c:246 frame #48: 0x000000010007c89a minirubyruby_exec_node(n=0x0000000100f21240) + 36 at eval.c:310 frame #49: 0x000000010007c86d minirubyruby_run_node(n=0x0000000100f21240) + 62 at eval.c:302 frame #50: 0x0000000100001399 minirubymain(argc=9, argv=0x00007fff5fbfdae0) + 113 at main.c:42 frame #51: 0x00007fff88eda5ad libdyld.dylibstart + 1 (lldb)

Revision 62023 - 01/24/2018 08:11 AM - naruse (Yui NARUSE)

merge revision(s) 61562,61563,61566,61568,61569: [Backport #14269]

```
fix SEGV touching uninitialized memory

This function can be called from boot_defclass().
No assumption can be made about object internals.

(lldb) run
Process 2386 launched: './miniruby' (x86_64)
Process 2386 stopped
* thread #1: tid = 0x13f3b6, 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variab
le.c:321, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x8)
    frame #0: 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variable.c:321
   318  VALUE
   319  rb_class_path_cached(VALUE klass)
   320  {
-> 321      st_table *ivtbl = RCLASS_IV_TBL(klass);
   322      st_data_t n;
   323
   324      if (!ivtbl) return Qnil;
(lldb) bt
* thread #1: tid = 0x13f3b6, 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variab
le.c:321, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x8)
  * frame #0: 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variable.c:321
    frame #1: 0x000000010009cbd0 miniruby`rb_raw_obj_info(buff="0x0000000100fa5798 [2    ] T_CLASS", buff_size
```

```
=256, obj=4311373720) + 1393 at gc.c:9341
    frame #2: 0x0000000010009cf16 miniruby`obj_info(obj=4311373720) + 98 at gc.c:9423
    frame #3: 0x0000000010008ca87 miniruby`newobj_init(klass=0, flags=66, v1=0, v2=0, v3=0, wb_protected=1, obj
space=0x00000001007cf280, obj=4311373720) + 338 at gc.c:1887
    frame #4: 0x0000000010008cd51 miniruby`newobj_of(klass=0, flags=66, v1=0, v2=0, v3=0, wb_protected=1) + 171
 at gc.c:1970
    frame #5: 0x0000000010008ce1b miniruby`rb_wb_protected_newobj_of(klass=0, flags=66) + 54 at gc.c:1990
    frame #6: 0x0000000100027563 miniruby`class_alloc(flags=2, klass=0) + 46 at class.c:165
    frame #7: 0x000000010002761a miniruby`rb_class_boot(super=0) + 35 at class.c:203
    frame #8: 0x0000000100028612 miniruby`boot_defclass(name="BasicObject", super=0) + 28 at class.c:537
    frame #9: 0x000000010002868b miniruby`Init_class_hierarchy + 26 at class.c:548
    frame #10: 0x00000001000efe69 miniruby`InitVM_Object + 9 at object.c:3892
    frame #11: 0x00000001000f138e miniruby`Init_Object + 57 at object.c:4122
    frame #12: 0x00000001000a59bd miniruby`rb_call_inits + 29 at inits.c:23
    frame #13: 0x0000000010007af30 miniruby`ruby_setup + 229 at eval.c:61
    frame #14: 0x0000000010007af7e miniruby`ruby_init + 13 at eval.c:78
    frame #15: 0x0000000100000c58 miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 88 at main.c:41
    frame #16: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)

fix SEGV touching uninitialized memory

This function can be called from InitVM_Object().
No assumption can be made about object internals.

(lldb) run
Process 10675 launched: './miniruby' (x86_64)
Process 10675 stopped
* thread #1: tid = 0x14252c, 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0    ] T_IM
EMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383, queue = 'com.apple.main-thread', stop reason =
EXC_BAD_ACCESS (code=1, address=0x0)
    frame #0: 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0    ] T_IMEMO ment", buff
_size=256, obj=4311487880) + 2489 at gc.c:9383
   9380                const rb_method_entry_t *me = &RANY(obj)->as.imemo.ment;
   9381                snprintf(buff, buff_size, "%s (called_id: %s, type: %s, alias: %d, owner: %s, defined_
class: %s)", buff,
   9382                         rb_id2name(me->called_id),
-> 9383                         method_type_name(me->def->type),
   9384                         me->def->alias_count,
   9385                         obj_info(me->owner),
   9386                         obj_info(me->defined_class));
(lldb) p *me
(rb_method_entry_t) $0 = {
  flags = 24602
  defined_class = 4311488400
  def = 0x0000000000000000
  called_id = 3057
  owner = 4311488400
}
(lldb) bt
* thread #1: tid = 0x14252c, 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0    ] T_IM
EMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383, queue = 'com.apple.main-thread', stop reason =
EXC_BAD_ACCESS (code=1, address=0x0)
  * frame #0: 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0    ] T_IMEMO ment", buff
_size=256, obj=4311487880) + 2489 at gc.c:9383
    frame #1: 0x00000001000b7cbf miniruby`obj_info(obj=4311487880) + 95 at gc.c:9423
    frame #2: 0x00000001000c16a8 miniruby`newobj_init(klass=4311488400, flags=24602, v1=0, v2=3057, v3=4311488
400, wb_protected=1, objspace=0x00000001007ee280, obj=4311487880) + 424 at gc.c:1887
    frame #3: 0x00000001000b4529 miniruby`newobj_of(klass=4311488400, flags=24602, v1=0, v2=3057, v3=431148840
0, wb_protected=1) + 217 at gc.c:1970
    frame #4: 0x00000001000b46ab miniruby`rb_imemo_new(type=imemo_ment, v1=0, v2=3057, v3=4311488400, v0=43114
88400) + 75 at gc.c:2017
    frame #5: 0x00000001002773b4 miniruby`rb_method_entry_alloc(called_id=3057, owner=4311488400, defined_clas
s=4311488400, def=0x0000000000000000) + 52 at vm_method.c:368
    frame #6: 0x0000000100277307 miniruby`rb_method_entry_create(called_id=3057, klass=4311488400, visi=METHOD
_VISI_PRIVATE, def=0x0000000000000000) + 71 at vm_method.c:389
    frame #7: 0x00000001002784c7 miniruby`rb_method_entry_make(klass=4311488400, mid=3057, defined_class=43114
88400, visi=METHOD_VISI_PRIVATE, type=VM_METHOD_TYPE_CFUNC, def=0x0000000000000000, original_id=3057, opts=0x0
0007fff5fbfd9e8) + 1207 at vm_method.c:594
    frame #8: 0x00000001002770f9 miniruby`rb_add_method(klass=4311488400, mid=3057, type=VM_METHOD_TYPE_CFUNC,
 opts=0x00007fff5fbfd9e8, visi=METHOD_VISI_PRIVATE) + 73 at vm_method.c:650
    frame #9: 0x000000010027708a miniruby`rb_add_method_cfunc(klass=4311488400, mid=3057, func=(miniruby`rb_ob
j_dummy at object.c:1125), argc=0, visi=METHOD_VISI_PRIVATE) + 138 at vm_method.c:137
    frame #10: 0x00000001000391e4 miniruby`rb_define_private_method(klass=4311488400, name="initialize", func=
(miniruby`rb_obj_dummy at object.c:1125), argc=0) + 68 at class.c:1529
```

```
    frame #11: 0x0000000010013f5bf miniruby`InitVM_Object + 47 at object.c:3905
    frame #12: 0x0000000100142ffd miniruby`Init_Object + 61 at object.c:4122
    frame #13: 0x00000001000d4edd miniruby`rb_call_inits + 29 at inits.c:23
    frame #14: 0x0000000010009fe66 miniruby`ruby_setup + 198 at eval.c:61
    frame #15: 0x0000000010009febd miniruby`ruby_init + 13 at eval.c:78
    frame #16: 0x0000000100000a4d miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 93 at main.c:41
    frame #17: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)

fix SEGV touching uninitialized local variable

This imemo_name is used uninitialized because the switch
above does not cover all possible imemo types.

(lldb) run
Process 26068 launched: './miniruby' (x86_64)
Process 26068 stopped
* thread #1: tid = 0x14ba96, 0x00007fff8a402132 libsystem_c.dylib`strlen + 18, queue = 'com.apple.main-thread'
, stop reason = EXC_BAD_ACCESS (code=1, address=0xfffffffffffffff0)
    frame #0: 0x00007fff8a402132 libsystem_c.dylib`strlen + 18
libsystem_c.dylib`strlen:
->  0x7fff8a402132 <+18>: pcmpeqb (%rdi), %xmm0
    0x7fff8a402136 <+22>: pmovmskb %xmm0, %esi
    0x7fff8a40213a <+26>: andq   $0xf, %rcx
    0x7fff8a40213e <+30>: orq    $-0x1, %rax
(lldb) bt
* thread #1: tid = 0x14ba96, 0x00007fff8a402132 libsystem_c.dylib`strlen + 18, queue = 'com.apple.main-thread'
, stop reason = EXC_BAD_ACCESS (code=1, address=0xfffffffffffffff0)
  * frame #0: 0x00007fff8a402132 libsystem_c.dylib`strlen + 18
    frame #1: 0x00000001001f1531 miniruby`BSD_vfprintf(fp=0x00007fff5fbfc9e0, fmt0="%s %s", ap=0x00007fff5fbfc
bf0) + 5873 at vsnprintf.c:1026
    frame #2: 0x00000001001ef213 miniruby`ruby_do_vsnprintf(str="0x0000000100f46450 [0    ] T_IMEMO", n=256, f
mt="%s %s", ap=0x00007fff5fbfcbf0) + 131 at sprintf.c:1285
    frame #3: 0x00000001001ef3ea miniruby`ruby_snprintf(str="0x0000000100f46450 [0    ] T_IMEMO", n=256, fmt="
%s %s") + 426 at sprintf.c:1300
    frame #4: 0x00000001000bdc61 miniruby`rb_raw_obj_info(buff="0x0000000100f46450 [0    ] T_IMEMO", buff_size
=256, obj=4310983760) + 2353 at gc.c:9376
    frame #5: 0x00000001000b7bff miniruby`obj_info(obj=4310983760) + 95 at gc.c:9428
    frame #6: 0x00000001000c1658 miniruby`newobj_init(klass=0, flags=36890, v1=0, v2=4303040512, v3=4310983800
, wb_protected=1, objspace=0x00000001007ee280, obj=4310983760) + 424 at gc.c:1887
    frame #7: 0x00000001000b4469 miniruby`newobj_of(klass=0, flags=36890, v1=0, v2=4303040512, v3=4310983800,
 wb_protected=1) + 217 at gc.c:1970
    frame #8: 0x00000001000b45eb miniruby`rb_imemo_new(type=imemo_ast, v1=0, v2=4303040512, v3=4310983800, v0=
0) + 75 at gc.c:2017
    frame #9: 0x0000000010011daed miniruby`rb_ast_new + 61 at node.c:1146
    frame #10: 0x0000000100160e15 miniruby`rb_parser_compile_file_path(vparser=4310984400, fname=4310984960, f
ile=4310984080, start=1) + 53 at parse.y:5776
    frame #11: 0x00000001001e18ea miniruby`load_file_internal(argp_v=140734799795024) + 1834 at ruby.c:1907
    frame #12: 0x00000001000a1bb5 miniruby`rb_ensure(b_proc=(miniruby`load_file_internal at ruby.c:1795), data
1=140734799795024, e_proc=(miniruby`restore_load_file at ruby.c:2007), data2=140734799795024) + 245 at eval.c:
1037
    frame #13: 0x00000001001df4a4 miniruby`load_file(parser=4310984400, fname=4310984960, f=4310984080, script
=1, opt=0x00007fff5fbfda28) + 100 at ruby.c:2026
    frame #14: 0x00000001001e084e miniruby`process_options(argc=0, argv=0x00007fff5fbfdc00, opt=0x00007fff5fbf
da28) + 3454 at ruby.c:1682
    frame #15: 0x00000001001dfaae miniruby`ruby_process_options(argc=2, argv=0x00007fff5fbfdbf0) + 238 at ruby
.c:2257
    frame #16: 0x0000000010009ff43 miniruby`ruby_options(argc=2, argv=0x00007fff5fbfdbf0) + 211 at eval.c:105
    frame #17: 0x0000000100000989 miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 105 at main.c:42
    frame #18: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb) up 4
frame #4: 0x00000001000bdc61 miniruby`rb_raw_obj_info(buff="0x0000000100f46450 [0    ] T_IMEMO", buff_size=256
, obj=4310983760) + 2353 at gc.c:9376
   9373 #undef IMEMO_NAME
   9374               default: UNREACHABLE;
   9375             }
-> 9376             snprintf(buff, buff_size, "%s %s", buff, imemo_name);
   9377
   9378             switch (imemo_type(obj)) {
   9379               case imemo_ment: {
(lldb) p imemo_name
(const char *) $0 = 0xffffffffffffffff
(lldb) p imemo_type(obj)
(imemo_type) $1 = imemo_ast
(lldb)
```

```
fix SEGV inspecting already freed objects

obj_info() assumes the given object is alive.  Passing freed
objects to it results in SEGV.

(lldb) run
Process 29718 launched: './miniruby' (x86_64)
Process 29718 stopped
* thread #1: tid = 0x3082c5, 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:26
9, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
    frame #0: 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:269
   266      }
   267      else {
   268          VM_ASSERT(RB_TYPE_P(pathobj, T_ARRAY));
-> 269          return RARRAY_AREF(pathobj, PATHOBJ_PATH);
   270      }
   271  }
   272
(lldb) bt
* thread #1: tid = 0x3082c5, 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:26
9, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
  * frame #0: 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:269
    frame #1: 0x00000001000c25ff miniruby`rb_iseq_path(iseq=0x000000010af34a20) + 32 at iseq.c:723
    frame #2: 0x000000010009db09 miniruby`rb_raw_iseq_info(buff="0x000000010af34a20 [1    ] T_IMEMO iseq", buf
f_size=256, iseq=0x000000010af34a20) + 69 at gc.c:9274
    frame #3: 0x000000010009e45a miniruby`rb_raw_obj_info(buff="0x000000010af34a20 [1    ] T_IMEMO iseq", buff
_size=256, obj=4478683680) + 2191 at gc.c:9397
    frame #4: 0x000000010009e4d5 miniruby`obj_info(obj=4478683680) + 98 at gc.c:9429
    frame #5: 0x0000000100091ae3 miniruby`gc_page_sweep(objspace=0x00000001007d3280, heap=0x00000001007d32a0,
 sweep_page=0x000000010ae07bc0) + 622 at gc.c:3529
    frame #6: 0x000000010009206a miniruby`gc_sweep_step(objspace=0x00000001007d3280, heap=0x00000001007d32a0)
 + 188 at gc.c:3705
    frame #7: 0x0000000100092254 miniruby`gc_sweep_continue(objspace=0x00000001007d3280, heap=0x00000001007d32
a0) + 133 at gc.c:3772
    frame #8: 0x000000010008d7f9 miniruby`heap_prepare(objspace=0x00000001007d3280, heap=0x00000001007d32a0) +
 48 at gc.c:1746
    frame #9: 0x000000010008d8a1 miniruby`heap_get_freeobj_from_next_freepage(objspace=0x00000001007d3280, hea
p=0x00000001007d32a0) + 37 at gc.c:1769
    frame #10: 0x000000010008d98d miniruby`heap_get_freeobj(objspace=0x00000001007d3280, heap=0x00000001007d32
a0) + 83 at gc.c:1803
    frame #11: 0x000000010008dcb0 miniruby`newobj_slowpath(klass=4334386280, flags=5, v1=0, v2=0, v3=0, objspa
ce=0x00000001007d3280, wb_protected=1) + 220 at gc.c:1930
    frame #12: 0x000000010008dd6c miniruby`newobj_slowpath_wb_protected(klass=4334386280, flags=5, v1=0, v2=0,
 v3=0, objspace=0x00000001007d3280) + 76 at gc.c:1942
    frame #13: 0x000000010008dea1 miniruby`newobj_of(klass=4334386280, flags=5, v1=0, v2=0, v3=0, wb_protected
=1) + 221 at gc.c:1974
    frame #14: 0x000000010008df39 miniruby`rb_wb_protected_newobj_of(klass=4334386280, flags=5) + 54 at gc.c:1
990
    frame #15: 0x0000000100195f7c miniruby`str_alloc(klass=4334386280) + 29 at string.c:692
    frame #16: 0x0000000100195fe9 miniruby`str_new0(klass=4334386280, ptr="gitm", len=4, termlen=1) + 73 at st
ring.c:714
    frame #17: 0x000000010019633e miniruby`rb_enc_str_new(ptr="gitm", len=4, enc=0x00000001025d50a0) + 81 at s
tring.c:766
    frame #18: 0x000000010010a80a miniruby`parser_str_new(p="gitm", n=4, enc=0x00000001025d50a0, func=66, enc0
=0x00000001025d50a0) + 50 at parse.y:5817
    frame #19: 0x000000010010ce1a miniruby`parser_parse_string(parser=0x00000001042ac5c0, quote=0x000000010460
c028) + 795 at parse.y:6675
    frame #20: 0x00000001001120bd miniruby`parser_yylex(parser=0x00000001042ac5c0) + 159 at parse.y:8281
    frame #21: 0x0000000100115068 miniruby`yylex(lval=0x00007fff5fbf9948, yylloc=0x00007fff5fbf9ab0, parser=0x
00000001042ac5c0) + 55 at parse.y:8931
    frame #22: 0x00000001000fc79f miniruby`ruby_yyparse(parser=0x00000001042ac5c0) + 1198 at parse.c:5798
    frame #23: 0x0000000100109f5a miniruby`yycompile0(arg=4364879296) + 317 at parse.y:5595
    frame #24: 0x0000000100214ef0 miniruby`rb_suppress_tracing(func=(miniruby`yycompile0 at parse.y:5565), arg
=4364879296) + 349 at vm_trace.c:397
    frame #25: 0x000000010010a1df miniruby`yycompile(parser=0x00000001042ac5c0, fname=4443743440, line=1) + 12
6 at parse.y:5637
    frame #26: 0x000000010010a4c1 miniruby`parser_compile_string(vparser=4443743480, fname=4443743440, s=44437
43520, line=1) + 191 at parse.y:5706
    frame #27: 0x000000010010a5b7 miniruby`rb_parser_compile_string_path(vparser=4443743480, f=4443743440, s=4
443743520, line=1) + 58 at parse.y:5730
    frame #28: 0x0000000100206025 miniruby`eval_make_iseq(src=4443743520, fname=4443743440, line=1, bind=0x000
0000000000000, base_block=0x00007fff5fbfb370) + 266 at vm_eval.c:1274
    frame #29: 0x0000000100206153 miniruby`eval_string_with_cref(self=4334412520, src=4443743520, cref=0x00000
00000000000, file=52, line=1) + 197 at vm_eval.c:1307
```

```
    frame #30: 0x0000000100206389 miniruby`rb_f_eval(argc=1, argv=0x0000000102400eb8, self=4334412520) + 219 a
t vm_eval.c:1382
    frame #31: 0x00000001001f247c miniruby`call_cfunc_m1(func=(miniruby`rb_f_eval at vm_eval.c:1364), recv=433
4412520, argc=1, argv=0x0000000102400eb8) + 47 at vm_insnhelper.c:1723
    frame #32: 0x00000001001f2f87 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x000000010
2500d80, calling=0x00007fff5fbfbf50, ci=0x000000010263f240, cc=0x0000000100749b50) + 386 at vm_insnhelper.c:19
18
    frame #33: 0x00000001001f30d6 miniruby`vm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500d80, ca
lling=0x00007fff5fbfbf50, ci=0x000000010263f240, cc=0x0000000100749b50) + 149 at vm_insnhelper.c:1934
    frame #34: 0x00000001001faf0e miniruby`vm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:
915
    frame #35: 0x0000000010020b75d miniruby`vm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771
    frame #36: 0x00000001002093f8 miniruby`invoke_block(ec=0x00000001007d3548, iseq=0x000000010252d7f0, self=4
334412520, captured=0x0000000102500df8, cref=0x0000000000000000, type=572653569, opt_pc=0) + 224 at vm.c:988
    frame #37: 0x0000000100209766 miniruby`invoke_iseq_block_from_c(ec=0x00000001007d3548, captured=0x00000001
02500df8, self=4334412520, argc=0, argv=0x0000000000000000, passed_block_handler=0, cref=0x0000000000000000, i
s_lambda=0) + 389 at vm.c:1040
    frame #38: 0x0000000100209824 miniruby`invoke_block_from_c_bh(ec=0x00000001007d3548, block_handler=4333768
185, argc=0, argv=0x0000000000000000, passed_block_handler=0, cref=0x0000000000000000, is_lambda=0, force_bloc
karg=0) + 138 at vm.c:1058
    frame #39: 0x00000001002099d0 miniruby`vm_yield(ec=0x00000001007d3548, argc=0, argv=0x0000000000000000) +
69 at vm.c:1103
    frame #40: 0x0000000100205623 miniruby`rb_yield_0(argc=0, argv=0x0000000000000000) + 40 at vm_eval.c:970
    frame #41: 0x0000000100205964 miniruby`loop_i + 19 at vm_eval.c:1049
    frame #42: 0x0000000010007db07 miniruby`rb_rescue2(b_proc=(miniruby`loop_i at vm_eval.c:1047), data1=0, r_p
roc=(miniruby`loop_stop at vm_eval.c:1056), data2=0) + 369 at eval.c:896
    frame #43: 0x0000000100205a2e miniruby`rb_f_loop(self=4334412520) + 121 at vm_eval.c:1100
    frame #44: 0x00000001001f24a7 miniruby`call_cfunc_0(func=(miniruby`rb_f_loop at vm_eval.c:1098), recv=4334
412520, argc=0, argv=0x0000000102400e80) + 41 at vm_insnhelper.c:1729
    frame #45: 0x00000001001f2f87 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x000000010
2500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 386 at vm_insnhelper.c:19
18
    frame #46: 0x00000001001f30d6 miniruby`vm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500de0, ca
lling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 149 at vm_insnhelper.c:1934
    frame #47: 0x00000001001f4319 miniruby`vm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500
de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 239 at vm_insnhelper.c:2232
    frame #48: 0x00000001001f4a2c miniruby`vm_call_method(ec=0x00000001007d3548, cfp=0x0000000102500de0, calli
ng=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 253 at vm_insnhelper.c:2366
    frame #49: 0x00000001001f4b7a miniruby`vm_call_general(ec=0x00000001007d3548, reg_cfp=0x0000000102500de0,
calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 59 at vm_insnhelper.c:2398
    frame #50: 0x00000001001fab2f miniruby`vm_exec_core(ec=0x00000001007d3548, initial=0) + 7480 at insns.def:
850
    frame #51: 0x0000000010020b75d miniruby`vm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771
    frame #52: 0x0000000010020c40f miniruby`rb_iseq_eval_main(iseq=0x000000010252dd90) + 52 at vm.c:2019
    frame #53: 0x0000000010007c768 miniruby`ruby_exec_internal(n=0x000000010252dd90) + 297 at eval.c:246
    frame #54: 0x0000000010007c88e miniruby`ruby_exec_node(n=0x000000010252dd90) + 36 at eval.c:310
    frame #55: 0x0000000010007c861 miniruby`ruby_run_node(n=0x000000010252dd90) + 62 at eval.c:302
    frame #56: 0x000000010000138d miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 113 at main.c:42
    frame #57: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb) p ((struct RVALUE*)pathobj)->as.basic
(RBasic) $0 = (flags = 0, klass = 4478683600)
(lldb)
```

```
fix SEGV inspecting uninitialized objects

obj_info() assumes the given object is alive.  OTOH
gc_writebarrier_incremental is called before or in middle of
object initialization.  Can casue SEGV.
```

```
(lldb) run
Process 48188 launched: './miniruby' (x86_64)
Process 48188 stopped
* thread #1: tid = 0x30fd53, 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at rub
y.h:2072, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=EXC_I386_GPFLT)
    frame #0: 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072
   2069 static inline const VALUE *
   2070 rb_array_const_ptr(VALUE a)
   2071 {
-> 2072     return FIX_CONST_VALUE_PTR((RBASIC(a)->flags & RARRAY_EMBED_FLAG) ?
   2073         RARRAY(a)->as.ary : RARRAY(a)->as.heap.ptr);
   2074 }
   2075
(lldb) bt
* thread #1: tid = 0x30fd53, 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at rub
y.h:2072, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=EXC_I386_GPFLT)
```

```
  * frame #0: 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072
    frame #1: 0x00000001000bfaab miniruby`pathobj_path(pathobj=5251291222225483145) + 70 at vm_core.h:269
    frame #2: 0x00000001000c25ff miniruby`rb_iseq_path(iseq=0x00000001025b71a8) + 32 at iseq.c:723
    frame #3: 0x0000000100009db09 miniruby`rb_raw_iseq_info(buff="0x00000001025b7158 [0    ] proc (Proc)", buff
_size=256, iseq=0x00000001025b71a8) + 69 at gc.c:9274
    frame #4: 0x0000000100009e1d5 miniruby`rb_raw_obj_info(buff="0x00000001025b7158 [0    ] proc (Proc)", buff_
size=256, obj=4334514520) + 1546 at gc.c:9351
    frame #5: 0x0000000100009e4d5 miniruby`obj_info(obj=4334514520) + 98 at gc.c:9429
    frame #6: 0x0000000100096658 miniruby`gc_writebarrier_incremental(a=4334514520, b=4334514600, objspace=0x0
0000001007d3280) + 61 at gc.c:5963
    frame #7: 0x00000001000968ca miniruby`rb_gc_writebarrier(a=4334514520, b=4334514600) + 127 at gc.c:6009
    frame #8: 0x00000001001eabe0 miniruby`rb_obj_written(a=4334514520, oldv=52, b=4334514600, filename="/Users
/urabe.shyouhei/data/src/pedantic/vm.c", line=821) + 72 at ruby.h:1472
    frame #9: 0x00000001001eac2c miniruby`rb_obj_write(a=4334514520, slot=0x0000000010259ff10, b=4334514600, fi
lename="/Users/urabe.shyouhei/data/src/pedantic/vm.c", line=821) + 70 at ruby.h:1489
    frame #10: 0x0000000100208b6f miniruby`vm_proc_create_from_captured(klass=4311027960, captured=0x000000010
2500338, block_type=block_type_ifunc, is_from_method='\0', is_lambda='\x01') + 137 at vm.c:821
    frame #11: 0x0000000100208e5c miniruby`rb_vm_make_proc_lambda(ec=0x00000001007d3548, captured=0x0000000102
500338, klass=4311027960, is_lambda='\x01') + 134 at vm.c:892
    frame #12: 0x0000000010011f08e miniruby`proc_new(klass=4311027960, is_lambda='\x01') + 445 at proc.c:752
    frame #13: 0x0000000010011f110 miniruby`rb_block_lambda + 27 at proc.c:808
    frame #14: 0x00000001001f24a7 miniruby`call_cfunc_0(func=(miniruby`rb_block_lambda at proc.c:807), recv=43
10991600, argc=0, argv=0x0000000000000000) + 41 at vm_insnhelper.c:1729
    frame #15: 0x00000001002033de miniruby`vm_call0_cfunc_with_frame(ec=0x00000001007d3548, calling=0x00007fff
5fbfb080, ci=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 370 at vm_eval.c:85
    frame #16: 0x00000001002034d9 miniruby`vm_call0_cfunc(ec=0x00000001007d3548, calling=0x00007fff5fbfb080, c
i=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 59 at vm_eval.c:100
    frame #17: 0x000000010020368f miniruby`vm_call0_body(ec=0x00000001007d3548, calling=0x00007fff5fbfb080, ci
=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 436 at vm_eval.c:131
    frame #18: 0x000000010020326a miniruby`vm_call0(ec=0x00000001007d3548, recv=4310991600, id=2993, argc=0, a
rgv=0x0000000000000000, me=0x0000000100f48110) + 142 at vm_eval.c:58
    frame #19: 0x0000000100203c60 miniruby`rb_call0(ec=0x00000001007d3548, recv=4310991600, mid=2993, argc=0,
 argv=0x0000000000000000, scope=CALL_FCALL, self=4334514640) + 166 at vm_eval.c:296
    frame #20: 0x0000000100204827 miniruby`rb_call(recv=4310991600, mid=2993, argc=0, argv=0x0000000000000000,
 scope=CALL_FCALL) + 84 at vm_eval.c:589
    frame #21: 0x000000010020518b miniruby`rb_funcallv(recv=4310991600, mid=2993, argc=0, argv=0x0000000000000
000) + 52 at vm_eval.c:815
    frame #22: 0x000000010012242e miniruby`mlambda(method=0) + 45 at proc.c:2661
    frame #23: 0x0000000100205bac miniruby`rb_iterate0(it_proc=(miniruby`mlambda at proc.c:2660), data1=0, ifu
nc=0x00000001025b71a8, ec=0x00000001007d3548) + 380 at vm_eval.c:1134
    frame #24: 0x0000000100205d16 miniruby`rb_iterate(it_proc=(miniruby`mlambda at proc.c:2660), data1=0, bl_p
roc=(miniruby`bmcall at proc.c:2666), data2=4334514640) + 88 at vm_eval.c:1166
    frame #25: 0x00000001001224c7 miniruby`method_to_proc(method=4334514640) + 43 at proc.c:2701
    frame #26: 0x00000001001f24a7 miniruby`call_cfunc_0(func=(miniruby`method_to_proc at proc.c:2688), recv=43
34514640, argc=0, argv=0x0000000102400568) + 41 at vm_insnhelper.c:1729
    frame #27: 0x00000001001f2f87 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x000000010
2500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 386 at vm_insnhelper.c:19
18
    frame #28: 0x00000001001f30d6 miniruby`vm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500350, ca
lling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 149 at vm_insnhelper.c:1934
    frame #29: 0x00000001001f4319 miniruby`vm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500
350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 239 at vm_insnhelper.c:2232
    frame #30: 0x00000001001f49a4 miniruby`vm_call_method(ec=0x00000001007d3548, cfp=0x0000000102500350, calli
ng=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 117 at vm_insnhelper.c:2355
    frame #31: 0x00000001001f4b7a miniruby`vm_call_general(ec=0x00000001007d3548, reg_cfp=0x0000000102500350,
 calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 59 at vm_insnhelper.c:2398
    frame #32: 0x00000001001faf0e miniruby`vm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:
915
    frame #33: 0x000000010020b75d miniruby`vm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771
    frame #34: 0x000000010020c3d1 miniruby`rb_iseq_eval(iseq=0x00000001007f8270) + 52 at vm.c:2008
    frame #35: 0x00000001000caa4a miniruby`rb_load_internal0(ec=0x00000001007d3548, fname=4310799960, wrap=0)
 + 631 at load.c:611
    frame #36: 0x00000001000cab36 miniruby`rb_load_internal(fname=4310799960, wrap=0) + 46 at load.c:642
    frame #37: 0x00000001000cae1d miniruby`rb_f_load(argc=1, argv=0x00000001024004b8) + 217 at load.c:710
    frame #38: 0x00000001001f247c miniruby`call_cfunc_m1(func=(miniruby`rb_f_load at load.c:695), recv=4311327
440, argc=1, argv=0x00000001024004b8) + 47 at vm_insnhelper.c:1723
    frame #39: 0x00000001001f2f87 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x000000010
25003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 386 at vm_insnhelper.c:19
18
    frame #40: 0x00000001001f30d6 miniruby`vm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x00000001025003e0, ca
lling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 149 at vm_insnhelper.c:1934
    frame #41: 0x00000001001f4319 miniruby`vm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500
3e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 239 at vm_insnhelper.c:2232
    frame #42: 0x00000001001f4a2c miniruby`vm_call_method(ec=0x00000001007d3548, cfp=0x00000001025003e0, calli
ng=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 253 at vm_insnhelper.c:2366
```

```
    frame #43: 0x00000001001f4b7a miniruby`vm_call_general(ec=0x00000001007d3548, reg_cfp=0x00000001025003e0,
calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 59 at vm_insnhelper.c:2398
    frame #44: 0x00000001001faf0e miniruby`vm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:
915
    frame #45: 0x000000010020b75d miniruby`vm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771
    frame #46: 0x000000010020c40f miniruby`rb_iseq_eval_main(iseq=0x0000000100f21240) + 52 at vm.c:2019
    frame #47: 0x0000000010007c774 miniruby`ruby_exec_internal(n=0x0000000100f21240) + 297 at eval.c:246
    frame #48: 0x0000000010007c89a miniruby`ruby_exec_node(n=0x0000000100f21240) + 36 at eval.c:310
    frame #49: 0x0000000010007c86d miniruby`ruby_run_node(n=0x0000000100f21240) + 62 at eval.c:302
    frame #50: 0x0000000100001399 miniruby`main(argc=9, argv=0x00007fff5fbfdae0) + 113 at main.c:42
    frame #51: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)
```

Revision 62095 - 01/29/2018 05:00 PM - naruse (Yui NARUSE)

merge revision(s) 61564,61565,61571: [Backport #14270]

```
fix SEGV touching uninitialized memory

This function can be called from Init_VM().
No assumption can be made about object internals.

(lldb) run
Process 15734 launched: './miniruby' (x86_64)
Process 15734 stopped
* thread #1: tid = 0x1441d4, 0x00000001000bdfcb miniruby`rb_raw_iseq_info(buff="0x0000000100f61f48 [0    ] T_I
MEMO iseq", buff_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273, queue = 'com.apple.main-thread', stop r
eason = EXC_BAD_ACCESS (code=1, address=0x50)
    frame #0: 0x00000001000bdfcb miniruby`rb_raw_iseq_info(buff="0x0000000100f61f48 [0    ] T_IMEMO iseq", buf
f_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273
   9270 static void
   9271 rb_raw_iseq_info(char *buff, const int buff_size, const rb_iseq_t *iseq)
   9272 {
-> 9273     if (iseq->body->location.label) {
   9274         VALUE path = rb_iseq_path(iseq);
   9275         snprintf(buff, buff_size, "%s %s@%s:%d", buff,
   9276                  RSTRING_PTR(iseq->body->location.label),
(lldb) p *iseq
(rb_iseq_t) $0 = {
  flags = 28698
  reserved1 = 0
  body = 0x0000000000000000
  aux = {
    compile_data = 0x0000000000000000
    loader = (obj = 0, index = 0)
    trace_events = 0
  }
}
(lldb) bt
* thread #1: tid = 0x1441d4, 0x00000001000bdfcb miniruby`rb_raw_iseq_info(buff="0x0000000100f61f48 [0    ] T_I
MEMO iseq", buff_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273, queue = 'com.apple.main-thread', stop r
eason = EXC_BAD_ACCESS (code=1, address=0x50)
  * frame #0: 0x00000001000bdfcb miniruby`rb_raw_iseq_info(buff="0x0000000100f61f48 [0    ] T_IMEMO iseq", buf
f_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273
    frame #1: 0x00000001000bde72 miniruby`rb_raw_obj_info(buff="0x0000000100f61f48 [0    ] T_IMEMO iseq", buff
_size=256, obj=4311097160) + 2786 at gc.c:9396
    frame #2: 0x00000001000b7c5f miniruby`obj_info(obj=4311097160) + 95 at gc.c:9428
    frame #3: 0x00000001000c16a8 miniruby`newobj_init(klass=0, flags=28698, v1=0, v2=0, v3=0, wb_protected=1,
objspace=0x00000001007ee280, obj=4311097160) + 424 at gc.c:1887
    frame #4: 0x00000001000b44c9 miniruby`newobj_of(klass=0, flags=28698, v1=0, v2=0, v3=0, wb_protected=1) +
217 at gc.c:1970
    frame #5: 0x00000001000b464b miniruby`rb_imemo_new(type=imemo_iseq, v1=0, v2=0, v3=0, v0=0) + 75 at gc.c:2
017
    frame #6: 0x00000001000fd914 miniruby`iseq_imemo_alloc + 36 at iseq.h:156
    frame #7: 0x00000001000f6e1d miniruby`iseq_alloc + 13 at iseq.c:211
    frame #8: 0x00000001000f6bf8 miniruby`rb_iseq_new_with_opt(node=0x0000000000000000, name=4311097200, path=
4311097200, realpath=8, first_lineno=1, parent=0x0000000000000000, type=ISEQ_TYPE_TOP, option=0x0000000100335c
30) + 56 at iseq.c:519
    frame #9: 0x00000001000f6bb6 miniruby`rb_iseq_new(node=0x0000000000000000, name=4311097200, path=431109720
0, realpath=8, parent=0x0000000000000000, type=ISEQ_TYPE_TOP) + 86 at iseq.c:480
    frame #10: 0x0000000100284bb0 miniruby`Init_VM + 1040 at vm.c:3022
    frame #11: 0x00000001000d4f7d miniruby`rb_call_inits + 189 at inits.c:55
    frame #12: 0x0000000010009fe06 miniruby`ruby_setup + 198 at eval.c:61
    frame #13: 0x0000000010009fe5d miniruby`ruby_init + 13 at eval.c:78
    frame #14: 0x00000001000009ed miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 93 at main.c:41
```

```
    frame #15: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)
```

fix SEGV touching uninitialized memory

This function can be called from rb_data_typed_object_zalloc().
No assumption can be made about object internals.

```
(lldb) run
Process 22135 launched: './miniruby' (x86_64)
Process 22135 stopped
* thread #1: tid = 0x14a3af, 0x0000000010008ac8a miniruby`vm_block_type(block=0x0000000000000000) + 12 at vm_co
re.h:1364, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x18)
    frame #0: 0x0000000010008ac8a miniruby`vm_block_type(block=0x0000000000000000) + 12 at vm_core.h:1364
   1361         break;
   1362     }
   1363 #endif
-> 1364     return block->type;
   1365 }
   1366
   1367 static inline void
(lldb) bt
* thread #1: tid = 0x14a3af, 0x0000000010008ac8a miniruby`vm_block_type(block=0x0000000000000000) + 12 at vm_co
re.h:1364, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x18)
  * frame #0: 0x0000000010008ac8a miniruby`vm_block_type(block=0x0000000000000000) + 12 at vm_core.h:1364
    frame #1: 0x0000000010008acdb miniruby`vm_block_iseq(block=0x0000000000000000) + 24 at vm_core.h:1399
    frame #2: 0x0000000010008acc1 miniruby`vm_proc_iseq(procval=4310866360) + 32 at vm_core.h:1387
    frame #3: 0x0000000010009cbed miniruby`rb_raw_obj_info(buff="0x0000000100f299b8 [0    ] proc (Proc)", buff_
size=256, obj=4310866360) + 1513 at gc.c:9349
    frame #4: 0x0000000010009cf01 miniruby`obj_info(obj=4310866360) + 98 at gc.c:9428
    frame #5: 0x0000000010008ca1b miniruby`newobj_init(klass=4311027960, flags=12, v1=4298186080, v2=1, v3=0, w
b_protected=32, objspace=0x00000001007cf280, obj=4310866360) + 338 at gc.c:1887
    frame #6: 0x0000000010008cce5 miniruby`newobj_of(klass=4311027960, flags=12, v1=4298186080, v2=1, v3=0, wb_
protected=32) + 171 at gc.c:1970
    frame #7: 0x0000000010008d01d miniruby`rb_data_typed_object_wrap(klass=4311027960, datap=0x0000000000000000
, type=0x0000000100311d60) + 133 at gc.c:2062
    frame #8: 0x0000000010008d04e miniruby`rb_data_typed_object_zalloc(klass=4311027960, size=40, type=0x000000
0100311d60) + 42 at gc.c:2073
    frame #9: 0x0000000010011b459 miniruby`rb_proc_alloc(klass=4311027960) + 36 at proc.c:113
    frame #10: 0x0000000100204d8e miniruby`vm_proc_create_from_captured(klass=4311027960, captured=0x000000010
25003f8, block_type=block_type_iseq, is_from_method='\0', is_lambda='\x01') + 44 at vm.c:814
    frame #11: 0x00000001002050d8 miniruby`rb_vm_make_proc_lambda(ec=0x00000001007cf548, captured=0x0000000102
5003f8, klass=4311027960, is_lambda='\x01') + 134 at vm.c:892
    frame #12: 0x0000000010011c0d2 miniruby`proc_new(klass=4311027960, is_lambda='\x01') + 445 at proc.c:752
    frame #13: 0x0000000010011c154 miniruby`rb_block_lambda + 27 at proc.c:808
    frame #14: 0x00000001001ee7e3 miniruby`call_cfunc_0(func=(miniruby`rb_block_lambda at proc.c:807), recv=43
10991600, argc=0, argv=0x0000000102400480) + 41 at vm_insnhelper.c:1729
    frame #15: 0x00000001001ef2c3 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007cf548, reg_cfp=0x000000010
25003e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x0000000010253e0f0) + 386 at vm_insnhelper.c:19
18
    frame #16: 0x00000001001ef412 miniruby`vm_call_cfunc(ec=0x00000001007cf548, reg_cfp=0x00000001025003e0, ca
lling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x0000000010253e0f0) + 149 at vm_insnhelper.c:1934
    frame #17: 0x00000001001f0655 miniruby`vm_call_method_each_type(ec=0x00000001007cf548, cfp=0x0000000102500
3e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x0000000010253e0f0) + 239 at vm_insnhelper.c:2232
    frame #18: 0x00000001001f0ce0 miniruby`vm_call_method(ec=0x00000001007cf548, cfp=0x00000001025003e0, calli
ng=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x0000000010253e0f0) + 117 at vm_insnhelper.c:2355
    frame #19: 0x00000001001f0eb6 miniruby`vm_call_general(ec=0x00000001007cf548, reg_cfp=0x00000001025003e0,
calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x0000000010253e0f0) + 59 at vm_insnhelper.c:2398
    frame #20: 0x00000001001f6e61 miniruby`vm_exec_core(ec=0x00000001007cf548, initial=0) + 7480 at insns.def:
850
    frame #21: 0x0000000100207995 miniruby`vm_exec(ec=0x00000001007cf548) + 230 at vm.c:1771
    frame #22: 0x0000000100208647 miniruby`rb_iseq_eval_main(iseq=0x0000000100f29fd0) + 52 at vm.c:2019
    frame #23: 0x0000000010007b750 miniruby`ruby_exec_internal(n=0x0000000100f29fd0) + 297 at eval.c:246
    frame #24: 0x0000000010007b876 miniruby`ruby_exec_node(n=0x0000000100f29fd0) + 36 at eval.c:310
    frame #25: 0x0000000010007b849 miniruby`ruby_run_node(n=0x0000000100f29fd0) + 62 at eval.c:302
    frame #26: 0x0000000100000c05 miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 113 at main.c:42
    frame #27: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)
```

check an existence of block.

```
* gc.c (rb_raw_obj_info): check block before using it.

* vm_core.h (vm_block_iseq): r61565 introduced NULL check but this
  check is only needed by `rb_raw_obj_info()` and it is called at GC
```

```
  debug mode. Above fix for `rb_raw_obj_info()` solves this problem and
  NULL check should not be needed any more.
```

Revision 62433 - 02/16/2018 02:45 PM - k0kubun (Takashi Kokubun)

mjit.c: fix deadlock on class serial increment

This is reported by @hasimo. Fixing a case like this:

#0 **lll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135**
#1 0x00007fc7bd824dbd in __GI_pthread_mutex_lock (mutex=mutex@entry=0x55946d294440) at ../nptl/pthread_mutex_lock.c:80 4
#2 0x000055946cec54d9 in rb_native_mutex_lock (lock=lock@entry=0x55946d294440) at thread_pthread.c:211
#3 0x000055946cde10ca in CRITICAL_SECTION_START (msg=0x55946cfb5423 "mjit_gc_start_hook", level=4) at mjit.c:392
#4 mjit_gc_start_hook () at mjit.c:412
#5 0x000055946cda0dfe in gc_enter (event=0x55946cfaf91e "gc_rest", objspace=0x55946da51760) at gc.c:6623
#6 gc_rest (objspace=objspace@entry=0x55946da51760) at gc.c:6515
#7 0x000055946cd9f1cf in gc_rest (objspace=0x55946da51760) at gc.c:7841
#8 objspace_malloc_increase (objspace=objspace@entry=0x55946da51760, new_size=, old_size=old_size@entry=0,
type=type@entry=MEMOP_TYPE_MALLOC, mem=0x7fc7a4439010) at gc.c:7842
#9 0x000055946cda1706 in objspace_malloc_fixup (size=, mem=0x7fc7a4439010, objspace=0x55946da51760) at gc.c:7910
#10 objspace_xmalloc0 (objspace=0x55946da51760, size=, size@entry=3145728) at gc.c:7939
#11 0x000055946cda3620 in ruby_xmalloc0 (size=3145728) at gc.c:8006
#12 ruby_xmalloc (size=size@entry=3145728) at gc.c:8015
#13 0x000055946ce93f4c in st_init_table_with_size (type=0x55946d28da30 , size=) at st.c:602
#14 0x000055946ce94287 in rebuild_table (tab=tab@entry=0x55946db669f0) at st.c:777
#15 0x000055946ce963f7 in rebuild_table_if_necessary (tab=0x55946db669f0) at st.c:1139
#16 st_add_direct_with_hash (hash=8577035585096733536, value=20, key=808451, tab=0x55946db669f0) at st.c:1207
#17 st_update (tab=0x55946db669f0, key=key@entry=808451, func=, arg=140726472841392) at st.c:1512
#18 0x000055946cda9e27 in tbl_update (optional_arg=, func=, key=, hash=) at hash.c:561
#19 rb_hash_aset (hash=94095983218480, key=key@entry=808451, val=val@entry=20) at hash.c:1654
#20 0x000055946cde243a in mjit_add_class_serial (class_serial=class_serial@entry=404225) at mjit.c:1414 3
#21 0x000055946cefcfab in rb_next_class_serial () at vm.c:321
#22 0x000055946cf48324 in class_alloc (klass=, flags=28) at class.c:178
#23 rb_include_class_new (module=module@entry=94096115733840, super=0) at class.c:820
#24 0x000055946cf487ac in include_modules_at (klass=klass@entry=94096135960920, c=, module=, module@entry=94096115734160,
search_super=search_super@entry=1) at class.c:913
#25 0x000055946cf48ac8 in rb_include_module (klass=94096135960920, module=module@entry=94096115734160) at class.c:870
#26 0x000055946cd84993 in rb_mod_append_features (module=94096115734160, include=) at eval.c:1178
#27 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6c9a20, cc=0x7ffd6f6c9ba0, argv=0x7ffd6f6c9ba0, calling=0x7ffd6f6c9a30,
ec=0x55946da519c8) at vm_eval.c:87
#28 vm_call0_cfunc (argv=0x7ffd6f6c9ba0, cc=0x7ffd6f6c9ba0, ci=0x7ffd6f6c9a20, calling=0x7ffd6f6c9a30, ec=0x55946da519c8) at vm_eval.c:102
#29 vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6c9ae0, ci=ci@entry=0x7ffd6f6c9ad0,
cc=cc@entry=0x7ffd6f6c9b00, argv=argv@entry=0x7ffd6f6c9ba0) at vm_eval.c:133
#30 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6c9ba0, argc=1, id=4849, recv=94096115734160, ec=0x55946da519c8) at vm_eval.c:60
#31 rb_call0 (ec=0x55946da519c8, recv=94096115734160, mid=4849, mid@entry=94096135960920, argc=argc@entry=1,
argv=argv@entry=0x7ffd6f6c9ba0, scope=scope@entry=CALL_FCALL, self=94096135960920) at vm_eval.c:302
#32 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x7ffd6f6c9ba0, argc=1, mid=94096135960920, recv=) at vm_eval.c:595
#33 rb_funcallv (recv=, mid=mid@entry=4849, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6c9ba0) at vm_eval.c:825
#34 0x000055946cd848a7 in rb_mod_include (argc=0, argv=0x7fc7bdb4fce8, module=94096135960920) at eval.c:1203
#35 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594710c06b0, cc=, calling=, reg_cfp=0x7fc7bdc4df00, ec=0x55946da519c8) at
vm_insnhelper.c:1928
#36 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4df00, calling=, ci=0x5594710c06b0, cc=) at vm_insnhelper.c:1944
#37 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at
/tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#38 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#39 0x000055946cf035fc in invoke_block (captured=0x7ffd6f6ca0a0, opt_pc=, type=, cref=0x559476c23930, self=94096135960920,
iseq=0x559471335ee0, ec=0x55946da519c8) at vm.c:1005
#40 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7ffd6f6ca0a0, self=94096135960920, argc=, argv=, passed_block_handler=0,
cref=0x559476c23930, is_lambda=0) at vm.c:1057
#41 0x000055946cf04520 in invoke_block_from_c_bh (ec=ec@entry=0x55946da519c8, block_handler=, argc=argc@entry=1,
argv=argv@entry=0x7ffd6f6ca108, cref=, is_lambda=, is_lambda@entry=0, force_blockarg=0, passed_block_handler=0) at vm.c:1075
#42 0x000055946cf04958 in vm_yield_with_cref (is_lambda=0, cref=, argv=0x7ffd6f6ca108, argc=1, ec=0x55946da519c8) at vm.c:1112
#43 yield_under (under=94096135960920, self=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108) at vm_eval.c:1572
#44 0x000055946cf04b12 in rb_mod_module_exec (argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108, mod=) at vm_eval.c:1770
#45 0x000055946ce00fc6 in rb_mod_initialize (module=94096135960920) at object.c:1978
#46 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6ca130, cc=0x7fc7bdb4fc98, argv=0x7fc7bdb4fc98, calling=0x7ffd6f6ca140,
ec=0x55946da519c8) at vm_eval.c:87
#47 vm_call0_cfunc (argv=0x7fc7bdb4fc98, cc=0x7fc7bdb4fc98, ci=0x7ffd6f6ca130, calling=0x7ffd6f6ca140, ec=0x55946da519c8) at vm_eval.c:102
#48 vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6ca1f0, ci=ci@entry=0x7ffd6f6ca1e0,
cc=cc@entry=0x7ffd6f6ca210, argv=argv@entry=0x7fc7bdb4fc98) at vm_eval.c:133
#49 0x000055946cf074b2 in vm_call0 (me=, argv=0x7fc7bdb4fc98, argc=0, id=3057, recv=94096135960920, ec=0x55946da519c8) at vm_eval.c:60
#50 rb_call0 (ec=0x55946da519c8, recv=recv@entry=94096135960920, mid=mid@entry=3057, argc=argc@entry=3057, argv=argv@entry=0x0,
scope=scope@entry=CALL_FCALL, self=94095983466120) at vm_eval.c:302
#51 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x0, argc=3057, mid=3057, recv=94096135960920) at vm_eval.c:595
#52 rb_funcallv (recv=recv@entry=94096135960920, mid=mid@entry=3057, argc=argc@entry=0, argv=argv@entry=0x7fc7bdb4fc98) at

vm_eval.c:825
#53 0x000055946cd89673 in rb_obj_call_init (obj=obj@entry=94096135960920, argc=argc@entry=0, argv=argv@entry=0x7fc7bdb4fc98) at eval.c:1590
#54 0x000055946ce048a1 in rb_class_s_new (argc=0, argv=0x7fc7bdb4fc98, klass=) at object.c:2153
#55 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x559471339c80, cc=, calling=, reg_cfp=0x7fc7bdc4dfa8, ec=0x55946da519c8) at vm_insnhelper.c:1928
#56 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4dfa8, calling=, ci=0x559471339c80, cc=) at vm_insnhelper.c:1944
#57 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#58 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#59 0x000055946cf03274 in invoke_bmethod (ec=ec@entry=0x55946da519c8, iseq=iseq@entry=0x55946f15cbc0, self=self@entry=94096115734640, me=me@entry=0x559475664e28, type=type@entry=572653825, opt_pc=0, captured=0x55947158e8a0) at vm.c:1026
#60 0x000055946cf03534 in invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x55947158e8a0, self=94096115734640, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=1) at vm.c:1060
#61 0x000055946cf036ce in invoke_block_from_c_proc (ec=, proc=, self=, argc=1, argv=, passed_block_handler=, is_lambda=) at vm.c:1150
#62 0x000055946cf03811 in vm_invoke_bmethod (block_handler=, argv=, argc=1, self=, proc=, ec=0x55946da519c8) at vm.c:1175
#63 vm_call_bmethod_body (ci=, cc=0x55946f625960, argv=, calling=0x7ffd6f6ca9b0, ec=0x55946da519c8) at vm_insnhelper.c:1971
#64 vm_call_bmethod (ec=0x55946da519c8, cfp=0x7fc7bdc4e0c0, calling=0x7ffd6f6ca9b0, ci=, cc=0x55946f625960) at vm_insnhelper.c:1988
#65 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e0c0, calling=, ci=, cc=) at vm_insnhelper.c:2417
#66 0x000055946cf0e63e in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:797
#67 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#68 0x000055946cf06767 in vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6cad00, ci=ci@entry=0x7ffd6f6cacf0, cc=cc@entry=0x7ffd6f6cad20, argv=argv@entry=0x7ffd6f6cadb0) at vm_eval.c:129
#69 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6cadb0, argc=1, id=3681, recv=94096115734640, ec=0x55946da519c8) at vm_eval.c:60
#70 rb_call0 (ec=0x55946da519c8, recv=recv@entry=94096115734640, mid=3681, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cad90, scope=scope@entry=CALL_FCALL, self=94095983452880) at vm_eval.c:302
#71 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x7ffd6f6cad90, argc=1, mid=, recv=94096115734640) at vm_eval.c:595
#72 rb_funcallv (recv=recv@entry=94096115734640, mid=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cadb0) at vm_eval.c:825
#73 0x000055946cf477d2 in rb_class_inherited (super=super@entry=94096115734640, klass=klass@entry=94096135961440) at class.c:625
#74 0x000055946cf0f796 in vm_declare_class (super=, cbase=94096094868200, flags=, id=847387) at vm_insnhelper.c:3134
#75 vm_define_class (super=, cbase=, flags=, id=847387) at vm_insnhelper.c:3167
#76 vm_find_or_create_class_by_id (super=, cbase=, flags=, id=847387) at vm_insnhelper.c:3196
#77 vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:685
#78 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#79 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x559471f1b548) at vm.c:2046
#80 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94096135872160, wrap=wrap@entry=0) at load.c:611
#81 0x000055946cdd41f1 in rb_require_internal (fname=94096135872400, fname@entry=94096135872440, safe=0) at load.c:992
#82 0x000055946cdd4493 in rb_require_safe (safe=, fname=94096135872440) at load.c:1038
#83 rb_f_require (obj=, fname=94096135872440) at load.c:820
#84 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594708700a0, cc=, calling=, reg_cfp=0x7fc7bdc4e168, ec=0x55946da519c8) at vm_insnhelper.c:1928
#85 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e168, calling=, ci=0x5594708700a0, cc=) at vm_insnhelper.c:1944
#86 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e168, calling=, ci=, cc=) at vm_insnhelper.c:2417
#87 0x000055946cf0e63e in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:797
#88 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#89 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4e490, opt_pc=, type=, cref=0x0, self=94096096426480, iseq=0x55946e49f0b8, ec=0x55946da519c8) at vm.c:1005
#90 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4e490, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#91 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#92 vm_yield (argc=1, argv=0x7ffd6f6cb938, ec=) at vm.c:1120
#93 rb_yield_0 (argv=0x7ffd6f6cb938, argc=1) at vm_eval.c:980
#94 rb_yield_1 (val=94096130169040) at vm_eval.c:986
#95 rb_yield (val=) at vm_eval.c:996
#96 0x000055946cf2113d in rb_ary_each (ary=94096076222560) at array.c:1820
#97 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946ed0d090, cc=, calling=, reg_cfp=0x7fc7bdc4e478, ec=0x55946da519c8) at vm_insnhelper.c:1928
#98 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e478, calling=, ci=0x55946ed0d090, cc=) at vm_insnhelper.c:1944
#99 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#100 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#101 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4e500, opt_pc=, type=, cref=0x0, self=94096096426480, iseq=0x55946e49f298, ec=0x55946da519c8) at vm.c:1005
#102 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4e500, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#103 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#104 vm_yield (argc=1, argv=0x7ffd6f6cbdf8, ec=) at vm.c:1120
#105 rb_yield_0 (argv=0x7ffd6f6cbdf8, argc=1) at vm_eval.c:980
#106 rb_yield_1 (val=94096095502480) at vm_eval.c:986

#107 rb_yield (val=) at vm_eval.c:996
#108 0x000055946cf2113d in rb_ary_each (ary=94096095328480) at array.c:1820
#109 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e8552a0, cc=, calling=, reg_cfp=0x7fc7bdc4e4e8, ec=0x55946da519c8) at vm_insnhelper.c:1928
#110 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e4e8, calling=, ci=0x55946e8552a0, cc=) at vm_insnhelper.c:1944
#111 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#112 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#113 0x000055946cf06767 in vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6cc2d0, ci=ci@entry=0x7ffd6f6cc2c0, cc=cc@entry=0x7ffd6f6cc2f0, argv=argv@entry=0x7ffd6f6cc390) at vm_eval.c:129
#114 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6cc390, argc=0, id=135807, recv=94096096186440, ec=0x55946da519c8) at vm_eval.c:60
#115 rb_call0 (ec=0x55946da519c8, recv=94096096186440, mid=135807, argc=, argv=argv@entry=0x8, scope=scope@entry=CALL_PUBLIC, self=94095993048320) at vm_eval.c:302
#116 0x000055946cf0a31a in rb_call (scope=CALL_PUBLIC, argv=0x8, argc=, mid=, recv=) at vm_eval.c:595
#117 rb_funcall_with_block (recv=, mid=, argc=argc@entry=0, argv=argv@entry=0x7ffd6f6cc390, passed_procval=passed_procval@entry=8) at vm_eval.c:857
#118 0x000055946ceb319c in rb_sym_proc_call (mid=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cc388, passed_proc=passed_proc@entry=8) at string.c:10480
#119 0x000055946cf0477c in vm_yield_with_symbol (block_handler=0, argv=0x7ffd6f6cc388, argc=1, symbol=, ec=) at vm_insnhelper.c:2573
#120 invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1084
#121 vm_yield (argc=1, argv=0x7ffd6f6cc388, ec=) at vm.c:1120
#122 rb_yield_0 (argv=0x7ffd6f6cc388, argc=1) at vm_eval.c:980
#123 rb_yield_1 (val=94096096186440) at vm_eval.c:986
#124 rb_yield (val=) at vm_eval.c:996
#125 0x000055946cf2113d in rb_ary_each (ary=94095993048320) at array.c:1820
#126 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594744d8280, cc=, calling=, reg_cfp=0x7fc7bdc4e590, ec=0x55946da519c8) at vm_insnhelper.c:1928
#127 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e590, calling=, ci=0x5594744d8280, cc=) at vm_insnhelper.c:1944
#128 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e590, calling=, ci=, cc=) at vm_insnhelper.c:2417
#129 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#130 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#131 0x000055946cf035fc in invoke_block (captured=0x7ffd6f6cc8e0, opt_pc=, type=, cref=0x559474604128, self=94096096426480, iseq=0x5594745027e8, ec=0x55946da519c8) at vm.c:1005
#132 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7ffd6f6cc8e0, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x559474604128, is_lambda=0) at vm.c:1057
#133 0x000055946cf04520 in invoke_block_from_c_bh (ec=ec@entry=0x55946da519c8, block_handler=, argc=argc@entry=1, argv=argv@entry=0x7fc7bdb4f7f8, cref=, is_lambda=, is_lambda@entry=0, force_blockarg=0, passed_block_handler=0) at vm.c:1075
#134 0x000055946cf04958 in vm_yield_with_cref (is_lambda=0, cref=, argv=0x7fc7bdb4f7f8, argc=1, ec=0x55946da519c8) at vm.c:1112
#135 yield_under (under=94096093646040, self=, argc=1, argv=0x7fc7bdb4f7f8) at vm_eval.c:1572
#136 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e5cd230, cc=, calling=, reg_cfp=0x7fc7bdc4e600, ec=0x55946da519c8) at vm_insnhelper.c:1928
#137 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e600, calling=, ci=0x55946e5cd230, cc=) at vm_insnhelper.c:1944
#138 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#139 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#140 0x000055946cf035fc in invoke_block (captured=0x5594744c4fb0, opt_pc=, type=, cref=0x0, self=94095990659320, iseq=0x55946e192aa0, ec=0x55946da519c8) at vm.c:1005
#141 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x5594744c4fb0, self=94095990659320, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#142 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#143 vm_yield (argc=1, argv=0x7ffd6f6ccdc8, ec=) at vm.c:1120
#144 rb_yield_0 (argv=0x7ffd6f6ccdc8, argc=1) at vm_eval.c:980
#145 rb_yield_1 (val=94096094867920) at vm_eval.c:986
#146 rb_yield (val=) at vm_eval.c:996
#147 0x000055946cf2113d in rb_ary_each (ary=94096094867160) at array.c:1820
#148 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6cce00, cc=0x7ffd6f6cce70, argv=0x7fc7bdb4f6b8, calling=0x7ffd6f6cce50, ec=0x55946da519c8) at vm_eval.c:87
#149 vm_call0_cfunc (argv=0x7fc7bdb4f6b8, cc=0x7ffd6f6cce70, ci=0x7ffd6f6cce00, calling=0x7ffd6f6cce50, ec=0x55946da519c8) at vm_eval.c:102
#150 vm_call0_body (ec=0x55946da519c8, calling=calling@entry=0x7ffd6f6ccec0, ci=ci@entry=0x7ffd6f6cceb0, cc=cc@entry=0x7ffd6f6ccee0, argv=0x7fc7bdb4f6b8) at vm_eval.c:133
#151 0x000055946cf06c50 in vm_call0 (me=, argv=, argc=, id=, recv=, ec=) at vm_eval.c:60
#152 rb_vm_call (ec=, recv=, id=, argc=, argv=, me=) at vm_eval.c:209
#153 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946dba2780, cc=, calling=, reg_cfp=0x7fc7bdc4e7c0, ec=0x55946da519c8) at vm_insnhelper.c:1928
#154 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e7c0, calling=, ci=0x55946dba2780, cc=) at vm_insnhelper.c:1944
#155 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e7c0, calling=, ci=, cc=) at vm_insnhelper.c:2417
#156 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#157 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#158 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946dfed3a8) at vm.c:2046
#159 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095988939080, wrap=wrap@entry=0) at load.c:611

#160 0x000055946cdd41f1 in rb_require_internal (fname=94095988939160, fname@entry=94095988939200, safe=0) at load.c:992
#161 0x000055946cdd4493 in rb_require_safe (safe=, fname=94095988939200) at load.c:1038
#162 rb_f_require (obj=, fname=94095988939200) at load.c:820
#163 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946dffa920, cc=, calling=, reg_cfp=0x7fc7bdc4e948, ec=0x55946da519c8) at vm_insnhelper.c:1928
#164 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e948, calling=, ci=0x55946dffa920, cc=) at vm_insnhelper.c:1944
#165 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e948, calling=, ci=, cc=) at vm_insnhelper.c:2417
#166 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#167 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#168 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946dfee438) at vm.c:2046
#169 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993039280, wrap=wrap@entry=0) at load.c:611
#170 0x000055946cdd41f1 in rb_require_internal (fname=fname@entry=94095993043840, safe=0) at load.c:992
#171 0x000055946cdd4493 in rb_require_safe (safe=, fname=94095993043840) at load.c:1038
#172 rb_f_require (obj=, fname=94095993043840) at load.c:820
#173 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946ea12160, cc=, calling=, reg_cfp=0x7fc7bdc4e9b8, ec=0x55946da519c8) at vm_insnhelper.c:1928
#174 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e9b8, calling=, ci=0x55946ea12160, cc=) at vm_insnhelper.c:1944
#175 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e9b8, calling=, ci=, cc=) at vm_insnhelper.c:2417
#176 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#177 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#178 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e3d7c48) at vm.c:2046
#179 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095988920840, wrap=) at load.c:611
#180 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095988920840) at load.c:642
#181 rb_f_load (argc=, argv=) at load.c:710
#182 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e7c96e0, cc=, calling=, reg_cfp=0x7fc7bdc4ea28, ec=0x55946da519c8) at vm_insnhelper.c:1928
#183 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ea28, calling=, ci=0x55946e7c96e0, cc=) at vm_insnhelper.c:1944
#184 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ea28, calling=, ci=, cc=) at vm_insnhelper.c:2417
#185 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#186 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#187 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4eae8, opt_pc=, type=, cref=0x0, self=94095998891400, iseq=0x55946e4dae10, ec=0x55946da519c8) at vm.c:1005
#188 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4eae8, self=94095998891400, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#189 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#190 vm_yield (argc=1, argv=0x7ffd6f6ce8e8, ec=) at vm.c:1120
#191 rb_yield_0 (argv=0x7ffd6f6ce8e8, argc=1) at vm_eval.c:980
#192 rb_yield_1 (val=94095988924840) at vm_eval.c:986
#193 rb_yield (val=) at vm_eval.c:996
#194 0x000055946cf2113d in rb_ary_each (ary=94095988920960) at array.c:1820
#195 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e421000, cc=, calling=, reg_cfp=0x7fc7bdc4ead0, ec=0x55946da519c8) at vm_insnhelper.c:1928
#196 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ead0, calling=, ci=0x55946e421000, cc=) at vm_insnhelper.c:1944
#197 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ead0, calling=, ci=, cc=) at vm_insnhelper.c:2417
#198 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#199 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#200 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e35f2c0) at vm.c:2046
#201 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095992643280, wrap=) at load.c:611
#202 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095992643280) at load.c:642
#203 rb_f_load (argc=, argv=) at load.c:710
#204 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e24bb20, cc=, calling=, reg_cfp=0x7fc7bdc4ec58, ec=0x55946da519c8) at vm_insnhelper.c:1928
#205 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ec58, calling=, ci=0x55946e24bb20, cc=) at vm_insnhelper.c:1944
#206 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ec58, calling=, ci=, cc=) at vm_insnhelper.c:2417
#207 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#208 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#209 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e3a7390) at vm.c:2046
#210 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993616120, wrap=) at load.c:611
#211 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095993616120) at load.c:642
#212 rb_f_load (argc=, argv=) at load.c:710
#213 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e6a00d0, cc=, calling=, reg_cfp=0x7fc7bdc4ecc8, ec=0x55946da519c8) at vm_insnhelper.c:1928
#214 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ecc8, calling=, ci=0x55946e6a00d0, cc=) at vm_insnhelper.c:1944
#215 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ecc8, calling=, ci=, cc=) at vm_insnhelper.c:2417
#216 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#217 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#218 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e470100) at vm.c:2046

#219 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993672200, wrap=) at load.c:611
#220 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095993672200) at load.c:642
#221 rb_f_load (argc=, argv=) at load.c:710
#222 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e3306a0, cc=, calling=, reg_cfp=0x7fc7bdc4efa0, ec=0x55946da519c8) at vm_insnhelper.c:1928
#223 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4efa0, calling=, ci=0x55946e3306a0, cc=) at vm_insnhelper.c:1944
#224 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4efa0, calling=, ci=, cc=) at vm_insnhelper.c:2417
#225 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#226 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#227 0x000055946cf119d5 in rb_iseq_eval_main (iseq=iseq@entry=0x55946e4bbba0) at vm.c:2057
#228 0x000055946cd83d54 in ruby_exec_internal (n=0x55946e4bbba0) at eval.c:247
#229 0x000055946cd87fdf in ruby_exec_node (n=0x55946e4bbba0) at eval.c:311
#230 ruby_run_node (n=) at eval.c:303
#231 0x000055946cd831bf in main (argc=22, argv=0x7ffd6f6d0148) at ./main.c:42

Revision 62433 - 02/16/2018 02:45 PM - k0kubun (Takashi Kokubun)

mjit.c: fix deadlock on class serial increment

This is reported by @hasimo. Fixing a case like this:

#0 **lll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135**
#1 0x00007fc7bd824dbd in __GI_pthread_mutex_lock (mutex=mutex@entry=0x55946d294440) at ../nptl/pthread_mutex_lock.c:80 4
#2 0x000055946cec54d9 in rb_native_mutex_lock (lock=lock@entry=0x55946d294440) at thread_pthread.c:211
#3 0x000055946cde10ca in CRITICAL_SECTION_START (msg=0x55946cfb5423 "mjit_gc_start_hook", level=4) at mjit.c:392
#4 mjit_gc_start_hook () at mjit.c:412
#5 0x000055946cda0dfe in gc_enter (event=0x55946cfaf91e "gc_rest", objspace=0x55946da51760) at gc.c:6623
#6 gc_rest (objspace=objspace@entry=0x55946da51760) at gc.c:6515
#7 0x000055946cd9f1cf in gc_rest (objspace=0x55946da51760) at gc.c:7841
#8 objspace_malloc_increase (objspace=objspace@entry=0x55946da51760, new_size=, old_size=old_size@entry=0, type=type@entry=MEMOP_TYPE_MALLOC, mem=0x7fc7a4439010) at gc.c:7842
#9 0x000055946cda1706 in objspace_malloc_fixup (size=, mem=0x7fc7a4439010, objspace=0x55946da51760) at gc.c:7910
#10 objspace_xmalloc0 (objspace=0x55946da51760, size=, size@entry=3145728) at gc.c:7939
#11 0x000055946cda3620 in ruby_xmalloc0 (size=3145728) at gc.c:8006
#12 ruby_xmalloc (size=size@entry=3145728) at gc.c:8015
#13 0x000055946ce93f4c in st_init_table_with_size (type=0x55946d28da30 , size=) at st.c:602
#14 0x000055946ce94287 in rebuild_table (tab=tab@entry=0x55946db669f0) at st.c:777
#15 0x000055946ce963f7 in rebuild_table_if_necessary (tab=0x55946db669f0) at st.c:1139
#16 st_add_direct_with_hash (hash=8577035585096733536, value=20, key=808451, tab=0x55946db669f0) at st.c:1207
#17 st_update (tab=0x55946db669f0, key=key@entry=808451, func=, arg=140726472841392) at st.c:1512
#18 0x000055946cda9e27 in tbl_update (optional_arg=, func=, key=, hash=) at hash.c:561
#19 rb_hash_aset (hash=94095983218480, key=key@entry=808451, val=val@entry=20) at hash.c:1654
#20 0x000055946cde243a in mjit_add_class_serial (class_serial=class_serial@entry=404225) at mjit.c:1414 3
#21 0x000055946cefcfab in rb_next_class_serial () at vm.c:321
#22 0x000055946cf48324 in class_alloc (klass=, flags=28) at class.c:178
#23 rb_include_class_new (module=module@entry=94096115733840, super=0) at class.c:820
#24 0x000055946cf487ac in include_modules_at (klass=klass@entry=94096135960920, c=, module=, module@entry=94096115734160, search_super=search_super@entry=1) at class.c:913
#25 0x000055946cf48ac8 in rb_include_module (klass=94096135960920, module=module@entry=94096115734160) at class.c:870
#26 0x000055946cd84993 in rb_mod_append_features (module=94096115734160, include=) at eval.c:1178
#27 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6c9a20, cc=0x7ffd6f6c9ba0, argv=0x7ffd6f6c9ba0, calling=0x7ffd6f6c9a30, ec=0x55946da519c8) at vm_eval.c:87
#28 vm_call0_cfunc (argv=0x7ffd6f6c9ba0, cc=0x7ffd6f6c9ba0, ci=0x7ffd6f6c9a20, calling=0x7ffd6f6c9a30, ec=0x55946da519c8) at vm_eval.c:102
#29 vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6c9ae0, ci=ci@entry=0x7ffd6f6c9ad0, cc=cc@entry=0x7ffd6f6c9b00, argv=argv@entry=0x7ffd6f6c9ba0) at vm_eval.c:133
#30 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6c9ba0, argc=1, id=4849, recv=94096115734160, ec=0x55946da519c8) at vm_eval.c:60
#31 rb_call0 (ec=0x55946da519c8, recv=94096115734160, mid=4849, mid@entry=94096135960920, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6c9ba0, scope=scope@entry=CALL_FCALL, self=94096135960920) at vm_eval.c:302
#32 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x7ffd6f6c9ba0, argc=1, mid=94096135960920, recv=) at vm_eval.c:595
#33 rb_funcallv (recv=, mid=mid@entry=4849, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6c9ba0) at vm_eval.c:825
#34 0x000055946cd848a7 in rb_mod_include (argc=0, argv=0x7fc7bdb4fce8, module=94096135960920) at eval.c:1203
#35 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594710c06b0, cc=, calling=, reg_cfp=0x7fc7bdc4df00, ec=0x55946da519c8) at vm_insnhelper.c:1928
#36 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4df00, calling=, ci=0x5594710c06b0, cc=) at vm_insnhelper.c:1944
#37 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#38 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#39 0x000055946cf035fc in invoke_block (captured=0x7ffd6f6ca0a0, opt_pc=, type=, cref=0x559476c23930, self=94096135960920, iseq=0x559471335ee0, ec=0x55946da519c8) at vm.c:1005
#40 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7ffd6f6ca0a0, self=94096135960920, argc=, argv=, passed_block_handler=0, cref=0x559476c23930, is_lambda=0) at vm.c:1057
#41 0x000055946cf04520 in invoke_block_from_c_bh (ec=ec@entry=0x55946da519c8, block_handler=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108, cref=, is_lambda=, is_lambda@entry=0, force_blockarg=0, passed_block_handler=0) at vm.c:1075
#42 0x000055946cf04958 in vm_yield_with_cref (is_lambda=0, cref=, argv=0x7ffd6f6ca108, argc=1, ec=0x55946da519c8) at vm.c:1112

#43 yield_under (under=94096135960920, self=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108) at vm_eval.c:1572
#44 0x000055946cf04b12 in rb_mod_module_exec (argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108, mod=) at vm_eval.c:1770
#45 0x000055946ce00fc6 in rb_mod_initialize (module=94096135960920) at object.c:1978
#46 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6ca130, cc=0x7fc7bdb4fc98, argv=0x7fc7bdb4fc98, calling=0x7ffd6f6ca140, ec=0x55946da519c8) at vm_eval.c:87
#47 vm_call0_cfunc (argv=0x7fc7bdb4fc98, cc=0x7fc7bdb4fc98, ci=0x7ffd6f6ca130, calling=0x7ffd6f6ca140, ec=0x55946da519c8) at vm_eval.c:102
#48 vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6ca1f0, ci=ci@entry=0x7ffd6f6ca1e0, cc=cc@entry=0x7ffd6f6ca210, argv=argv@entry=0x7fc7bdb4fc98) at vm_eval.c:133
#49 0x000055946cf074b2 in vm_call0 (me=, argv=0x7fc7bdb4fc98, argc=0, id=3057, recv=94096135960920, ec=0x55946da519c8) at vm_eval.c:60
#50 rb_call0 (ec=0x55946da519c8, recv=recv@entry=94096135960920, mid=mid@entry=3057, argc=argc@entry=3057, argv=argv@entry=0x0, scope=scope@entry=CALL_FCALL, self=94095983466120) at vm_eval.c:302
#51 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x0, argc=3057, mid=3057, recv=94096135960920) at vm_eval.c:595
#52 rb_funcallv (recv=recv@entry=94096135960920, mid=mid@entry=3057, argc=argc@entry=0, argv=argv@entry=0x7fc7bdb4fc98) at vm_eval.c:825
#53 0x000055946cd89673 in rb_obj_call_init (obj=obj@entry=94096135960920, argc=argc@entry=0, argv=argv@entry=0x7fc7bdb4fc98) at eval.c:1590
#54 0x000055946ce048a1 in rb_class_s_new (argc=0, argv=0x7fc7bdb4fc98, klass=) at object.c:2153
#55 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x559471339c80, cc=, calling=, reg_cfp=0x7fc7bdc4dfa8, ec=0x55946da519c8) at vm_insnhelper.c:1928
#56 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4dfa8, calling=, ci=0x559471339c80, cc=) at vm_insnhelper.c:1944
#57 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#58 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#59 0x000055946cf03274 in invoke_bmethod (ec=ec@entry=0x55946da519c8, iseq=iseq@entry=0x55946f15cbc0, self=self@entry=94096115734640, me=me@entry=0x559475664e28, type=type@entry=572653825, opt_pc=0, captured=0x55947158e8a0) at vm.c:1026
#60 0x000055946cf03534 in invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x55947158e8a0, self=94096115734640, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=1) at vm.c:1060
#61 0x000055946cf036ce in invoke_block_from_c_proc (ec=, proc=, self=, argc=1, argv=, passed_block_handler=, is_lambda=) at vm.c:1150
#62 0x000055946cf03811 in vm_invoke_bmethod (block_handler=, argv=, argc=1, self=, proc=, ec=0x55946da519c8) at vm.c:1175
#63 vm_call_bmethod_body (ci=, cc=0x55946f625960, argv=, calling=0x7ffd6f6ca9b0, ec=0x55946da519c8) at vm_insnhelper.c:1971
#64 vm_call_bmethod (ec=0x55946da519c8, cfp=0x7fc7bdc4e0c0, calling=0x7ffd6f6ca9b0, ci=, cc=0x55946f625960) at vm_insnhelper.c:1988
#65 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e0c0, calling=, ci=, cc=) at vm_insnhelper.c:2417
#66 0x000055946cf0e63e in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:797
#67 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#68 0x000055946cf06767 in vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6cad00, ci=ci@entry=0x7ffd6f6cacf0, cc=cc@entry=0x7ffd6f6cad20, argv=argv@entry=0x7ffd6f6cadb0) at vm_eval.c:129
#69 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6cadb0, argc=1, id=3681, recv=94096115734640, ec=0x55946da519c8) at vm_eval.c:60
#70 rb_call0 (ec=0x55946da519c8, recv=recv@entry=94096115734640, mid=3681, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cad90, scope=scope@entry=CALL_FCALL, self=94095983452880) at vm_eval.c:302
#71 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x7ffd6f6cad90, argc=1, mid=, recv=94096115734640) at vm_eval.c:595
#72 rb_funcallv (recv=recv@entry=94096115734640, mid=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cadb0) at vm_eval.c:825
#73 0x000055946cf477d2 in rb_class_inherited (super=super@entry=94096115734640, klass=klass@entry=94096135961440) at class.c:625
#74 0x000055946cf0f796 in vm_declare_class (super=, cbase=94096094868200, flags=, id=847387) at vm_insnhelper.c:3134
#75 vm_define_class (super=, cbase=, flags=, id=847387) at vm_insnhelper.c:3167
#76 vm_find_or_create_class_by_id (super=, cbase=, flags=, id=847387) at vm_insnhelper.c:3196
#77 vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:685
#78 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#79 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x559471f1b548) at vm.c:2046
#80 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94096135872160, wrap=wrap@entry=0) at load.c:611
#81 0x000055946cdd41f1 in rb_require_internal (fname=94096135872400, fname@entry=94096135872440, safe=0) at load.c:992
#82 0x000055946cdd4493 in rb_require_safe (safe=, fname=94096135872440) at load.c:1038
#83 rb_f_require (obj=, fname=94096135872440) at load.c:820
#84 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594708700a0, cc=, calling=, reg_cfp=0x7fc7bdc4e168, ec=0x55946da519c8) at vm_insnhelper.c:1928
#85 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e168, calling=, ci=0x5594708700a0, cc=) at vm_insnhelper.c:1944
#86 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e168, calling=, ci=, cc=) at vm_insnhelper.c:2417
#87 0x000055946cf0e63e in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:797
#88 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#89 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4e490, opt_pc=, type=, cref=0x0, self=94096096426480, iseq=0x55946e49f0b8, ec=0x55946da519c8) at vm.c:1005
#90 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4e490, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#91 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#92 vm_yield (argc=1, argv=0x7ffd6f6cb938, ec=) at vm.c:1120
#93 rb_yield_0 (argv=0x7ffd6f6cb938, argc=1) at vm_eval.c:980
#94 rb_yield_1 (val=94096130169040) at vm_eval.c:986
#95 rb_yield (val=) at vm_eval.c:996
#96 0x000055946cf2113d in rb_ary_each (ary=94096076222560) at array.c:1820
#97 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946ed0d090, cc=, calling=, reg_cfp=0x7fc7bdc4e478, ec=0x55946da519c8) at vm_insnhelper.c:1928

#98 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e478, calling=, ci=0x55946ed0d090, cc=) at vm_insnhelper.c:1944
#99 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#100 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#101 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4e500, opt_pc=, type=, cref=0x0, self=94096096426480, iseq=0x55946e49f298, ec=0x55946da519c8) at vm.c:1005
#102 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4e500, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#103 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#104 vm_yield (argc=1, argv=0x7ffd6f6cbdf8, ec=) at vm.c:1120
#105 rb_yield_0 (argv=0x7ffd6f6cbdf8, argc=1) at vm_eval.c:980
#106 rb_yield_1 (val=94096095502480) at vm_eval.c:986
#107 rb_yield (val=) at vm_eval.c:996
#108 0x000055946cf2113d in rb_ary_each (ary=94096095328480) at array.c:1820
#109 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e8552a0, cc=, calling=, reg_cfp=0x7fc7bdc4e4e8, ec=0x55946da519c8) at vm_insnhelper.c:1928
#110 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e4e8, calling=, ci=0x55946e8552a0, cc=) at vm_insnhelper.c:1944
#111 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#112 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#113 0x000055946cf06767 in vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6cc2d0, ci=ci@entry=0x7ffd6f6cc2c0, cc=cc@entry=0x7ffd6f6cc2f0, argv=argv@entry=0x7ffd6f6cc390) at vm_eval.c:129
#114 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6cc390, argc=0, id=135807, recv=94096096186440, ec=0x55946da519c8) at vm_eval.c:60
#115 rb_call0 (ec=0x55946da519c8, recv=94096096186440, mid=135807, argc=, argv=argv@entry=0x8, scope=scope@entry=CALL_PUBLIC, self=94095993048320) at vm_eval.c:302
#116 0x000055946cf0a31a in rb_call (scope=CALL_PUBLIC, argv=0x8, argc=, mid=, recv=) at vm_eval.c:595
#117 rb_funcall_with_block (recv=, mid=, argc=argc@entry=0, argv=argv@entry=0x7ffd6f6cc390, passed_procval=passed_procval@entry=8) at vm_eval.c:857
#118 0x000055946ceb319c in rb_sym_proc_call (mid=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cc388, passed_proc=passed_proc@entry=8) at string.c:10480
#119 0x000055946cf0477c in vm_yield_with_symbol (block_handler=0, argv=0x7ffd6f6cc388, argc=1, symbol=, ec=) at vm_insnhelper.c:2573
#120 invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1084
#121 vm_yield (argc=1, argv=0x7ffd6f6cc388, ec=) at vm.c:1120
#122 rb_yield_0 (argv=0x7ffd6f6cc388, argc=1) at vm_eval.c:980
#123 rb_yield_1 (val=94096096186440) at vm_eval.c:986
#124 rb_yield (val=) at vm_eval.c:996
#125 0x000055946cf2113d in rb_ary_each (ary=94095993048320) at array.c:1820
#126 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594744d8280, cc=, calling=, reg_cfp=0x7fc7bdc4e590, ec=0x55946da519c8) at vm_insnhelper.c:1928
#127 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e590, calling=, ci=0x5594744d8280, cc=) at vm_insnhelper.c:1944
#128 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e590, calling=, ci=, cc=) at vm_insnhelper.c:2417
#129 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#130 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#131 0x000055946cf035fc in invoke_block (captured=0x7ffd6f6cc8e0, opt_pc=, type=, cref=0x559474604128, self=94096096426480, iseq=0x5594745027e8, ec=0x55946da519c8) at vm.c:1005
#132 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7ffd6f6cc8e0, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x559474604128, is_lambda=0) at vm.c:1057
#133 0x000055946cf04520 in invoke_block_from_c_bh (ec=ec@entry=0x55946da519c8, block_handler=, argc=argc@entry=1, argv=argv@entry=0x7fc7bdb4f7f8, cref=, is_lambda=, is_lambda@entry=0, force_blockarg=0, passed_block_handler=0) at vm.c:1075
#134 0x000055946cf04958 in vm_yield_with_cref (is_lambda=0, cref=, argv=0x7fc7bdb4f7f8, argc=1, ec=0x55946da519c8) at vm.c:1112
#135 yield_under (under=94096093646040, self=, argc=1, argv=0x7fc7bdb4f7f8) at vm_eval.c:1572
#136 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e5cd230, cc=, calling=, reg_cfp=0x7fc7bdc4e600, ec=0x55946da519c8) at vm_insnhelper.c:1928
#137 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e600, calling=, ci=0x55946e5cd230, cc=) at vm_insnhelper.c:1944
#138 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#139 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#140 0x000055946cf035fc in invoke_block (captured=0x5594744c4fb0, opt_pc=, type=, cref=0x0, self=94095990659320, iseq=0x55946e192aa0, ec=0x55946da519c8) at vm.c:1005
#141 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x5594744c4fb0, self=94095990659320, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#142 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#143 vm_yield (argc=1, argv=0x7ffd6f6ccdc8, ec=) at vm.c:1120
#144 rb_yield_0 (argv=0x7ffd6f6ccdc8, argc=1) at vm_eval.c:980
#145 rb_yield_1 (val=94096094867920) at vm_eval.c:986
#146 rb_yield (val=) at vm_eval.c:996
#147 0x000055946cf2113d in rb_ary_each (ary=94096094867160) at array.c:1820
#148 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6cce00, cc=0x7ffd6f6cce70, argv=0x7fc7bdb4f6b8, calling=0x7ffd6f6cce50, ec=0x55946da519c8) at vm_eval.c:87
#149 vm_call0_cfunc (argv=0x7fc7bdb4f6b8, cc=0x7ffd6f6cce70, ci=0x7ffd6f6cce00, calling=0x7ffd6f6cce50, ec=0x55946da519c8) at vm_eval.c:102
#150 vm_call0_body (ec=0x55946da519c8, calling=calling@entry=0x7ffd6f6ccec0, ci=ci@entry=0x7ffd6f6cceb0, cc=cc@entry=0x7ffd6f6ccee0,

argv=0x7fc7bdb4f6b8) at vm_eval.c:133
#151 0x000055946cf06c50 in vm_call0 (me=, argv=, argc=, id=, recv=, ec=) at vm_eval.c:60
#152 rb_vm_call (ec=, recv=, id=, argc=, argv=, me=) at vm_eval.c:209
#153 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946dba2780, cc=, calling=, reg_cfp=0x7fc7bdc4e7c0, ec=0x55946da519c8) at vm_insnhelper.c:1928
#154 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e7c0, calling=, ci=0x55946dba2780, cc=) at vm_insnhelper.c:1944
#155 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e7c0, calling=, ci=, cc=) at vm_insnhelper.c:2417
#156 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#157 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#158 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946dfed3a8) at vm.c:2046
#159 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095988939080, wrap=wrap@entry=0) at load.c:611
#160 0x000055946cdd41f1 in rb_require_internal (fname=94095988939160, fname@entry=94095988939200, safe=0) at load.c:992
#161 0x000055946cdd4493 in rb_require_safe (safe=, fname=94095988939200) at load.c:1038
#162 rb_f_require (obj=, fname=94095988939200) at load.c:820
#163 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946dffa920, cc=, calling=, reg_cfp=0x7fc7bdc4e948, ec=0x55946da519c8) at vm_insnhelper.c:1928
#164 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e948, calling=, ci=0x55946dffa920, cc=) at vm_insnhelper.c:1944
#165 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e948, calling=, ci=, cc=) at vm_insnhelper.c:2417
#166 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#167 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#168 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946dfee438) at vm.c:2046
#169 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993039280, wrap=wrap@entry=0) at load.c:611
#170 0x000055946cdd41f1 in rb_require_internal (fname=fname@entry=94095993043840, safe=0) at load.c:992
#171 0x000055946cdd4493 in rb_require_safe (safe=, fname=94095993043840) at load.c:1038
#172 rb_f_require (obj=, fname=94095993043840) at load.c:820
#173 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946ea12160, cc=, calling=, reg_cfp=0x7fc7bdc4e9b8, ec=0x55946da519c8) at vm_insnhelper.c:1928
#174 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e9b8, calling=, ci=0x55946ea12160, cc=) at vm_insnhelper.c:1944
#175 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e9b8, calling=, ci=, cc=) at vm_insnhelper.c:2417
#176 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#177 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#178 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e3d7c48) at vm.c:2046
#179 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095988920840, wrap=) at load.c:611
#180 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095988920840) at load.c:642
#181 rb_f_load (argc=, argv=) at load.c:710
#182 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e7c96e0, cc=, calling=, reg_cfp=0x7fc7bdc4ea28, ec=0x55946da519c8) at vm_insnhelper.c:1928
#183 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ea28, calling=, ci=0x55946e7c96e0, cc=) at vm_insnhelper.c:1944
#184 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ea28, calling=, ci=, cc=) at vm_insnhelper.c:2417
#185 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#186 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#187 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4eae8, opt_pc=, type=, cref=0x0, self=94095998891400, iseq=0x55946e4dae10, ec=0x55946da519c8) at vm.c:1005
#188 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4eae8, self=94095998891400, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#189 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#190 vm_yield (argc=1, argv=0x7ffd6f6ce8e8, ec=) at vm.c:1120
#191 rb_yield_0 (argv=0x7ffd6f6ce8e8, argc=1) at vm_eval.c:980
#192 rb_yield_1 (val=94095988924840) at vm_eval.c:986
#193 rb_yield (val=) at vm_eval.c:996
#194 0x000055946cf2113d in rb_ary_each (ary=94095988920960) at array.c:1820
#195 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e421000, cc=, calling=, reg_cfp=0x7fc7bdc4ead0, ec=0x55946da519c8) at vm_insnhelper.c:1928
#196 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ead0, calling=, ci=0x55946e421000, cc=) at vm_insnhelper.c:1944
#197 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ead0, calling=, ci=, cc=) at vm_insnhelper.c:2417
#198 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#199 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#200 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e35f2c0) at vm.c:2046
#201 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095992643280, wrap=) at load.c:611
#202 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095992643280) at load.c:642
#203 rb_f_load (argc=, argv=) at load.c:710
#204 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e24bb20, cc=, calling=, reg_cfp=0x7fc7bdc4ec58, ec=0x55946da519c8) at vm_insnhelper.c:1928
#205 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ec58, calling=, ci=0x55946e24bb20, cc=) at vm_insnhelper.c:1944
#206 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ec58, calling=, ci=, cc=) at vm_insnhelper.c:2417
#207 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779

```
#208 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#209 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e3a7390) at vm.c:2046
#210 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993616120, wrap=) at load.c:611
#211 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095993616120) at load.c:642
#212 rb_f_load (argc=, argv=) at load.c:710
#213 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e6a00d0, cc=, calling=, reg_cfp=0x7fc7bdc4ecc8, ec=0x55946da519c8) at
vm_insnhelper.c:1928
#214 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ecc8, calling=, ci=0x55946e6a00d0, cc=) at vm_insnhelper.c:1944
#215 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ecc8, calling=, ci=, cc=) at vm_insnhelper.c:2417
#216 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at
/tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#217 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#218 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e470100) at vm.c:2046
#219 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993672200, wrap=) at load.c:611
#220 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095993672200) at load.c:642
#221 rb_f_load (argc=, argv=) at load.c:710
#222 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e3306a0, cc=, calling=, reg_cfp=0x7fc7bdc4efa0, ec=0x55946da519c8) at
vm_insnhelper.c:1928
#223 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4efa0, calling=, ci=0x55946e3306a0, cc=) at vm_insnhelper.c:1944
#224 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4efa0, calling=, ci=, cc=) at vm_insnhelper.c:2417
#225 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at
/tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#226 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#227 0x000055946cf119d5 in rb_iseq_eval_main (iseq=iseq@entry=0x55946e4bbba0) at vm.c:2057
#228 0x000055946cd83d54 in ruby_exec_internal (n=0x55946e4bbba0) at eval.c:247
#229 0x000055946cd87fdf in ruby_exec_node (n=0x55946e4bbba0) at eval.c:311
#230 ruby_run_node (n=) at eval.c:303
#231 0x000055946cd831bf in main (argc=22, argv=0x7ffd6f6d0148) at ./main.c:42
```

Revision 62850 - 03/19/2018 06:16 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 61562,61563,61566,61568,61569: [Backport #14269]

```
fix SEGV touching uninitialized memory

This function can be called from boot_defclass().
No assumption can be made about object internals.

(lldb) run
Process 2386 launched: './miniruby' (x86_64)
Process 2386 stopped
* thread #1: tid = 0x13f3b6, 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variab
le.c:321, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x8)
    frame #0: 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variable.c:321
   318  VALUE
   319  rb_class_path_cached(VALUE klass)
   320  {
-> 321      st_table *ivtbl = RCLASS_IV_TBL(klass);
   322      st_data_t n;
   323
   324      if (!ivtbl) return Qnil;
(lldb) bt
* thread #1: tid = 0x13f3b6, 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variab
le.c:321, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x8)
  * frame #0: 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variable.c:321
    frame #1: 0x000000010009cbd0 miniruby`rb_raw_obj_info(buff="0x0000000100fa5798 [2    ] T_CLASS", buff_size
=256, obj=4311373720) + 1393 at gc.c:9341
    frame #2: 0x000000010009cf16 miniruby`obj_info(obj=4311373720) + 98 at gc.c:9423
    frame #3: 0x000000010008ca87 miniruby`newobj_init(klass=0, flags=66, v1=0, v2=0, v3=0, wb_protected=1, obj
space=0x00000001007cf280, obj=4311373720) + 338 at gc.c:1887
    frame #4: 0x000000010008cd51 miniruby`newobj_of(klass=0, flags=66, v1=0, v2=0, v3=0, wb_protected=1) + 171
 at gc.c:1970
    frame #5: 0x000000010008ce1b miniruby`rb_wb_protected_newobj_of(klass=0, flags=66) + 54 at gc.c:1990
    frame #6: 0x0000000100027563 miniruby`class_alloc(flags=2, klass=0) + 46 at class.c:165
    frame #7: 0x000000010002761a miniruby`rb_class_boot(super=0) + 35 at class.c:203
    frame #8: 0x0000000100028612 miniruby`boot_defclass(name="BasicObject", super=0) + 28 at class.c:537
    frame #9: 0x000000010002868b miniruby`Init_class_hierarchy + 26 at class.c:548
    frame #10: 0x00000001000efe69 miniruby`InitVM_Object + 9 at object.c:3892
    frame #11: 0x00000001000f138e miniruby`Init_Object + 57 at object.c:4122
    frame #12: 0x00000001000a59bd miniruby`rb_call_inits + 29 at inits.c:23
    frame #13: 0x000000010007af30 miniruby`ruby_setup + 229 at eval.c:61
    frame #14: 0x000000010007af7e miniruby`ruby_init + 13 at eval.c:78
    frame #15: 0x0000000100000c58 miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 88 at main.c:41
    frame #16: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)
```

```
fix SEGV touching uninitialized memory

This function can be called from InitVM_Object().
No assumption can be made about object internals.

(lldb) run
Process 10675 launched: './miniruby' (x86_64)
Process 10675 stopped
* thread #1: tid = 0x14252c, 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0    ] T_IM
EMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383, queue = 'com.apple.main-thread', stop reason =
EXC_BAD_ACCESS (code=1, address=0x0)
    frame #0: 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0    ] T_IMEMO ment", buff
_size=256, obj=4311487880) + 2489 at gc.c:9383
   9380                const rb_method_entry_t *me = &RANY(obj)->as.imemo.ment;
   9381                snprintf(buff, buff_size, "%s (called_id: %s, type: %s, alias: %d, owner: %s, defined_
class: %s)", buff,
   9382                         rb_id2name(me->called_id),
-> 9383                         method_type_name(me->def->type),
   9384                         me->def->alias_count,
   9385                         obj_info(me->owner),
   9386                         obj_info(me->defined_class));
(lldb) p *me
(rb_method_entry_t) $0 = {
  flags = 24602
  defined_class = 4311488400
  def = 0x0000000000000000
  called_id = 3057
  owner = 4311488400
}
(lldb) bt
* thread #1: tid = 0x14252c, 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0    ] T_IM
EMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383, queue = 'com.apple.main-thread', stop reason =
EXC_BAD_ACCESS (code=1, address=0x0)
  * frame #0: 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0    ] T_IMEMO ment", buff
_size=256, obj=4311487880) + 2489 at gc.c:9383
    frame #1: 0x00000001000b7cbf miniruby`obj_info(obj=4311487880) + 95 at gc.c:9423
    frame #2: 0x00000001000c16a8 miniruby`newobj_init(klass=4311488400, flags=24602, v1=0, v2=3057, v3=4311488
400, wb_protected=1, objspace=0x00000001007ee280, obj=4311487880) + 424 at gc.c:1887
    frame #3: 0x00000001000b4529 miniruby`newobj_of(klass=4311488400, flags=24602, v1=0, v2=3057, v3=431148840
0, wb_protected=1) + 217 at gc.c:1970
    frame #4: 0x00000001000b46ab miniruby`rb_imemo_new(type=imemo_ment, v1=0, v2=3057, v3=4311488400, v0=43114
88400) + 75 at gc.c:2017
    frame #5: 0x00000001002773b4 miniruby`rb_method_entry_alloc(called_id=3057, owner=4311488400, defined_clas
s=4311488400, def=0x0000000000000000) + 52 at vm_method.c:368
    frame #6: 0x0000000100277307 miniruby`rb_method_entry_create(called_id=3057, klass=4311488400, visi=METHOD
_VISI_PRIVATE, def=0x0000000000000000) + 71 at vm_method.c:389
    frame #7: 0x00000001002784c7 miniruby`rb_method_entry_make(klass=4311488400, mid=3057, defined_class=43114
88400, visi=METHOD_VISI_PRIVATE, type=VM_METHOD_TYPE_CFUNC, def=0x0000000000000000, original_id=3057, opts=0x0
0007fff5fbfd9e8) + 1207 at vm_method.c:594
    frame #8: 0x00000001002770f9 miniruby`rb_add_method(klass=4311488400, mid=3057, type=VM_METHOD_TYPE_CFUNC,
 opts=0x00007fff5fbfd9e8, visi=METHOD_VISI_PRIVATE) + 73 at vm_method.c:650
    frame #9: 0x000000010027708a miniruby`rb_add_method_cfunc(klass=4311488400, mid=3057, func=(miniruby`rb_ob
j_dummy at object.c:1125), argc=0, visi=METHOD_VISI_PRIVATE) + 138 at vm_method.c:137
    frame #10: 0x00000001000391e4 miniruby`rb_define_private_method(klass=4311488400, name="initialize", func=
(miniruby`rb_obj_dummy at object.c:1125), argc=0) + 68 at class.c:1529
    frame #11: 0x0000000100013f5bf miniruby`InitVM_Object + 47 at object.c:3905
    frame #12: 0x0000000100142ffd miniruby`Init_Object + 61 at object.c:4122
    frame #13: 0x00000001000d4edd miniruby`rb_call_inits + 29 at inits.c:23
    frame #14: 0x0000000100009fe66 miniruby`ruby_setup + 198 at eval.c:61
    frame #15: 0x0000000100009febd miniruby`ruby_init + 13 at eval.c:78
    frame #16: 0x0000000100000a4d miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 93 at main.c:41
    frame #17: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)

fix SEGV touching uninitialized local variable

This imemo_name is used uninitialized because the switch
above does not cover all possible imemo types.

(lldb) run
Process 26068 launched: './miniruby' (x86_64)
Process 26068 stopped
* thread #1: tid = 0x14ba96, 0x00007fff8a402132 libsystem_c.dylib`strlen + 18, queue = 'com.apple.main-thread'
, stop reason = EXC_BAD_ACCESS (code=1, address=0xfffffffffffffff0)
```

```
    frame #0: 0x00007fff8a402132 libsystem_c.dylib`strlen + 18
libsystem_c.dylib`strlen:
->  0x7fff8a402132 <+18>: pcmpeqb (%rdi), %xmm0
    0x7fff8a402136 <+22>: pmovmskb %xmm0, %esi
    0x7fff8a40213a <+26>: andq   $0xf, %rcx
    0x7fff8a40213e <+30>: orq    $-0x1, %rax
(lldb) bt
* thread #1: tid = 0x14ba96, 0x00007fff8a402132 libsystem_c.dylib`strlen + 18, queue = 'com.apple.main-thread'
, stop reason = EXC_BAD_ACCESS (code=1, address=0xffffffffffffffff0)
  * frame #0: 0x00007fff8a402132 libsystem_c.dylib`strlen + 18
    frame #1: 0x00000001001f1531 miniruby`BSD_vfprintf(fp=0x00007fff5fbfc9e0, fmt0="%s %s", ap=0x00007fff5fbfc
bf0) + 5873 at vsnprintf.c:1026
    frame #2: 0x00000001001ef213 miniruby`ruby_do_vsnprintf(str="0x0000000100f46450 [0    ] T_IMEMO", n=256, f
mt="%s %s", ap=0x00007fff5fbfcbf0) + 131 at sprintf.c:1285
    frame #3: 0x00000001001ef3ea miniruby`ruby_snprintf(str="0x0000000100f46450 [0    ] T_IMEMO", n=256, fmt="
%s %s") + 426 at sprintf.c:1300
    frame #4: 0x00000001000bdc61 miniruby`rb_raw_obj_info(buff="0x0000000100f46450 [0    ] T_IMEMO", buff_size
=256, obj=4310983760) + 2353 at gc.c:9376
    frame #5: 0x00000001000b7bff miniruby`obj_info(obj=4310983760) + 95 at gc.c:9428
    frame #6: 0x00000001000c1658 miniruby`newobj_init(klass=0, flags=36890, v1=0, v2=4303040512, v3=4310983800
, wb_protected=1, objspace=0x000000001007ee280, obj=4310983760) + 424 at gc.c:1887
    frame #7: 0x00000001000b4469 miniruby`newobj_of(klass=0, flags=36890, v1=0, v2=4303040512, v3=4310983800,
 wb_protected=1) + 217 at gc.c:1970
    frame #8: 0x00000001000b45eb miniruby`rb_imemo_new(type=imemo_ast, v1=0, v2=4303040512, v3=4310983800, v0=
0) + 75 at gc.c:2017
    frame #9: 0x0000000010011daed miniruby`rb_ast_new + 61 at node.c:1146
    frame #10: 0x0000000100160e15 miniruby`rb_parser_compile_file_path(vparser=4310984400, fname=4310984960, f
ile=4310984080, start=1) + 53 at parse.y:5776
    frame #11: 0x00000001001e18ea miniruby`load_file_internal(argp_v=140734799795024) + 1834 at ruby.c:1907
    frame #12: 0x00000001000a1bb5 miniruby`rb_ensure(b_proc=(miniruby`load_file_internal at ruby.c:1795), data
1=140734799795024, e_proc=(miniruby`restore_load_file at ruby.c:2007), data2=140734799795024) + 245 at eval.c:
1037
    frame #13: 0x00000001001df4a4 miniruby`load_file(parser=4310984400, fname=4310984960, f=4310984080, script
=1, opt=0x00007fff5fbfda28) + 100 at ruby.c:2026
    frame #14: 0x00000001001e084e miniruby`process_options(argc=0, argv=0x00007fff5fbfdc00, opt=0x00007fff5fbf
da28) + 3454 at ruby.c:1682
    frame #15: 0x00000001001dfaae miniruby`ruby_process_options(argc=2, argv=0x00007fff5fbfdbf0) + 238 at ruby
.c:2257
    frame #16: 0x0000000010009ff43 miniruby`ruby_options(argc=2, argv=0x00007fff5fbfdbf0) + 211 at eval.c:105
    frame #17: 0x0000000100000989 miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 105 at main.c:42
    frame #18: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb) up 4
frame #4: 0x00000001000bdc61 miniruby`rb_raw_obj_info(buff="0x0000000100f46450 [0    ] T_IMEMO", buff_size=256
, obj=4310983760) + 2353 at gc.c:9376
   9373 #undef IMEMO_NAME
   9374               default: UNREACHABLE;
   9375             }
-> 9376             snprintf(buff, buff_size, "%s %s", buff, imemo_name);
   9377
   9378             switch (imemo_type(obj)) {
   9379               case imemo_ment: {
(lldb) p imemo_name
(const char *) $0 = 0xffffffffffffffff
(lldb) p imemo_type(obj)
(imemo_type) $1 = imemo_ast
(lldb)

fix SEGV inspecting already freed objects

obj_info() assumes the given object is alive.  Passing freed
objects to it results in SEGV.

(lldb) run
Process 29718 launched: './miniruby' (x86_64)
Process 29718 stopped
* thread #1: tid = 0x3082c5, 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:26
9, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
    frame #0: 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:269
   266      }
   267      else {
   268          VM_ASSERT(RB_TYPE_P(pathobj, T_ARRAY));
-> 269          return RARRAY_AREF(pathobj, PATHOBJ_PATH);
   270      }
   271  }
   272
```

```
(lldb) bt
* thread #1: tid = 0x3082c5, 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:269, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
  * frame #0: 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:269
    frame #1: 0x00000001000c25ff miniruby`rb_iseq_path(iseq=0x000000010af34a20) + 32 at iseq.c:723
    frame #2: 0x0000000010009db09 miniruby`rb_raw_iseq_info(buff="0x000000010af34a20 [1    ] T_IMEMO iseq", buf_size=256, iseq=0x000000010af34a20) + 69 at gc.c:9274
    frame #3: 0x0000000010009e45a miniruby`rb_raw_obj_info(buff="0x000000010af34a20 [1    ] T_IMEMO iseq", buff_size=256, obj=4478683680) + 2191 at gc.c:9397
    frame #4: 0x0000000010009e4d5 miniruby`obj_info(obj=4478683680) + 98 at gc.c:9429
    frame #5: 0x0000000100091ae3 miniruby`gc_page_sweep(objspace=0x00000001007d3280, heap=0x00000001007d32a0, sweep_page=0x000000010ae07bc0) + 622 at gc.c:3529
    frame #6: 0x0000000010009206a miniruby`gc_sweep_step(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 188 at gc.c:3705
    frame #7: 0x0000000100092254 miniruby`gc_sweep_continue(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 133 at gc.c:3772
    frame #8: 0x0000000010008d7f9 miniruby`heap_prepare(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 48 at gc.c:1746
    frame #9: 0x0000000010008d8a1 miniruby`heap_get_freeobj_from_next_freepage(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 37 at gc.c:1769
    frame #10: 0x0000000010008d98d miniruby`heap_get_freeobj(objspace=0x00000001007d3280, heap=0x00000001007d32a0) + 83 at gc.c:1803
    frame #11: 0x0000000010008dcb0 miniruby`newobj_slowpath(klass=4334386280, flags=5, v1=0, v2=0, v3=0, objspace=0x00000001007d3280, wb_protected=1) + 220 at gc.c:1930
    frame #12: 0x0000000010008dd6c miniruby`newobj_slowpath_wb_protected(klass=4334386280, flags=5, v1=0, v2=0, v3=0, objspace=0x00000001007d3280) + 76 at gc.c:1942
    frame #13: 0x0000000010008dea1 miniruby`newobj_of(klass=4334386280, flags=5, v1=0, v2=0, v3=0, wb_protected=1) + 221 at gc.c:1974
    frame #14: 0x0000000010008df39 miniruby`rb_wb_protected_newobj_of(klass=4334386280, flags=5) + 54 at gc.c:1990
    frame #15: 0x00000000100195f7c miniruby`str_alloc(klass=4334386280) + 29 at string.c:692
    frame #16: 0x00000000100195fe9 miniruby`str_new0(klass=4334386280, ptr="gitm", len=4, termlen=1) + 73 at string.c:714
    frame #17: 0x0000000010019633e miniruby`rb_enc_str_new(ptr="gitm", len=4, enc=0x00000001025d50a0) + 81 at string.c:766
    frame #18: 0x0000000010010a80a miniruby`parser_str_new(p="gitm", n=4, enc=0x00000001025d50a0, func=66, enc0=0x00000001025d50a0) + 50 at parse.y:5817
    frame #19: 0x0000000010010ce1a miniruby`parser_parse_string(parser=0x00000001042ac5c0, quote=0x000000010460c028) + 795 at parse.y:6675
    frame #20: 0x00000001001120bd miniruby`parser_yylex(parser=0x00000001042ac5c0) + 159 at parse.y:8281
    frame #21: 0x0000000100115068 miniruby`yylex(lval=0x00007fff5fbf9948, yylloc=0x00007fff5fbf9ab0, parser=0x00000001042ac5c0) + 55 at parse.y:8931
    frame #22: 0x00000001000fc79f miniruby`ruby_yyparse(parser=0x00000001042ac5c0) + 1198 at parse.c:5798
    frame #23: 0x00000000100109f5a miniruby`yycompile0(arg=4364879296) + 317 at parse.y:5595
    frame #24: 0x0000000100214ef0 miniruby`rb_suppress_tracing(func=(miniruby`yycompile0 at parse.y:5565), arg=4364879296) + 349 at vm_trace.c:397
    frame #25: 0x0000000010010a1df miniruby`yycompile(parser=0x00000001042ac5c0, fname=4443743440, line=1) + 126 at parse.y:5637
    frame #26: 0x0000000010010a4c1 miniruby`parser_compile_string(vparser=4443743480, fname=4443743440, s=4443743520, line=1) + 191 at parse.y:5706
    frame #27: 0x0000000010010a5b7 miniruby`rb_parser_compile_string_path(vparser=4443743480, f=4443743440, s=4443743520, line=1) + 58 at parse.y:5730
    frame #28: 0x0000000100206025 miniruby`eval_make_iseq(src=4443743520, fname=4443743440, line=1, bind=0x0000000000000000, base_block=0x00007fff5fbfb370) + 266 at vm_eval.c:1274
    frame #29: 0x0000000100206153 miniruby`eval_string_with_cref(self=4334412520, src=4443743520, cref=0x0000000000000000, file=52, line=1) + 197 at vm_eval.c:1307
    frame #30: 0x0000000100206389 miniruby`rb_f_eval(argc=1, argv=0x0000000102400eb8, self=4334412520) + 219 at vm_eval.c:1382
    frame #31: 0x00000001001f247c miniruby`call_cfunc_m1(func=(miniruby`rb_f_eval at vm_eval.c:1364), recv=4334412520, argc=1, argv=0x0000000102400eb8) + 47 at vm_insnhelper.c:1723
    frame #32: 0x00000001001f2f87 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x0000000102500d80, calling=0x00007fff5fbfbf50, ci=0x000000010263f240, cc=0x0000000100749b50) + 386 at vm_insnhelper.c:1918
    frame #33: 0x00000001001f30d6 miniruby`vm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500d80, calling=0x00007fff5fbfbf50, ci=0x000000010263f240, cc=0x0000000100749b50) + 149 at vm_insnhelper.c:1934
    frame #34: 0x00000001001faf0e miniruby`vm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:915
    frame #35: 0x0000000010020b75d miniruby`vm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771
    frame #36: 0x00000001002093f8 miniruby`invoke_block(ec=0x00000001007d3548, iseq=0x000000010252d7f0, self=4334412520, captured=0x0000000102500df8, cref=0x0000000000000000, type=572653569, opt_pc=0) + 224 at vm.c:988
    frame #37: 0x0000000100209766 miniruby`invoke_iseq_block_from_c(ec=0x00000001007d3548, captured=0x0000000102500df8, self=4334412520, argc=0, argv=0x0000000000000000, passed_block_handler=0, cref=0x0000000000000000, is_lambda=0) + 389 at vm.c:1040
    frame #38: 0x0000000100209824 miniruby`invoke_block_from_c_bh(ec=0x00000001007d3548, block_handler=4333768185, argc=0, argv=0x0000000000000000, passed_block_handler=0, cref=0x0000000000000000, is_lambda=0, force_bloc
```

```
karg=0) + 138 at vm.c:1058
    frame #39: 0x00000001002099d0 miniruby`vm_yield(ec=0x00000001007d3548, argc=0, argv=0x0000000000000000) +
69 at vm.c:1103
    frame #40: 0x0000000100205623 miniruby`rb_yield_0(argc=0, argv=0x0000000000000000) + 40 at vm_eval.c:970
    frame #41: 0x0000000100205964 miniruby`loop_i + 19 at vm_eval.c:1049
    frame #42: 0x000000010007db07 miniruby`rb_rescue2(b_proc=(miniruby`loop_i at vm_eval.c:1047), data1=0, r_p
roc=(miniruby`loop_stop at vm_eval.c:1056), data2=0) + 369 at eval.c:896
    frame #43: 0x0000000100205a2e miniruby`rb_f_loop(self=4334412520) + 121 at vm_eval.c:1100
    frame #44: 0x00000001001f24a7 miniruby`call_cfunc_0(func=(miniruby`rb_f_loop at vm_eval.c:1098), recv=4334
412520, argc=0, argv=0x0000000102400e80) + 41 at vm_insnhelper.c:1729
    frame #45: 0x00000001001f2f87 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x000000010
2500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 386 at vm_insnhelper.c:19
18
    frame #46: 0x00000001001f30d6 miniruby`vm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500de0, ca
lling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 149 at vm_insnhelper.c:1934
    frame #47: 0x00000001001f4319 miniruby`vm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500
de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 239 at vm_insnhelper.c:2232
    frame #48: 0x00000001001f4a2c miniruby`vm_call_method(ec=0x00000001007d3548, cfp=0x0000000102500de0, calli
ng=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 253 at vm_insnhelper.c:2366
    frame #49: 0x00000001001f4b7a miniruby`vm_call_general(ec=0x00000001007d3548, reg_cfp=0x0000000102500de0,
calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 59 at vm_insnhelper.c:2398
    frame #50: 0x00000001001fab2f miniruby`vm_exec_core(ec=0x00000001007d3548, initial=0) + 7480 at insns.def:
850
    frame #51: 0x000000010020b75d miniruby`vm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771
    frame #52: 0x000000010020c40f miniruby`rb_iseq_eval_main(iseq=0x000000010252dd90) + 52 at vm.c:2019
    frame #53: 0x000000010007c768 miniruby`ruby_exec_internal(n=0x000000010252dd90) + 297 at eval.c:246
    frame #54: 0x000000010007c88e miniruby`ruby_exec_node(n=0x000000010252dd90) + 36 at eval.c:310
    frame #55: 0x000000010007c861 miniruby`ruby_run_node(n=0x000000010252dd90) + 62 at eval.c:302
    frame #56: 0x000000010000138d miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 113 at main.c:42
    frame #57: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb) p ((struct RVALUE*)pathobj)->as.basic
(RBasic) $0 = (flags = 0, klass = 4478683600)
(lldb)
```

```
fix SEGV inspecting uninitialized objects

obj_info() assumes the given object is alive.  OTOH
gc_writebarrier_incremental is called before or in middle of
object initialization.  Can casue SEGV.
```

```
(lldb) run
Process 48188 launched: './miniruby' (x86_64)
Process 48188 stopped
* thread #1: tid = 0x30fd53, 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at rub
y.h:2072, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=EXC_I386_GPFLT)
    frame #0: 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072
   2069 static inline const VALUE *
   2070 rb_array_const_ptr(VALUE a)
   2071 {
-> 2072     return FIX_CONST_VALUE_PTR((RBASIC(a)->flags & RARRAY_EMBED_FLAG) ?
   2073         RARRAY(a)->as.ary : RARRAY(a)->as.heap.ptr);
   2074 }
   2075
(lldb) bt
* thread #1: tid = 0x30fd53, 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at rub
y.h:2072, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=EXC_I386_GPFLT)
  * frame #0: 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072
    frame #1: 0x00000001000bfaab miniruby`pathobj_path(pathobj=5251291222225483145) + 70 at vm_core.h:269
    frame #2: 0x00000001000c25ff miniruby`rb_iseq_path(iseq=0x00000001025b71a8) + 32 at iseq.c:723
    frame #3: 0x000000010009db09 miniruby`rb_raw_iseq_info(buff="0x00000001025b7158 [0    ] proc (Proc)", buff
_size=256, iseq=0x00000001025b71a8) + 69 at gc.c:9274
    frame #4: 0x000000010009e1d5 miniruby`rb_raw_obj_info(buff="0x00000001025b7158 [0    ] proc (Proc)", buff_
size=256, obj=4334514520) + 1546 at gc.c:9351
    frame #5: 0x000000010009e4d5 miniruby`obj_info(obj=4334514520) + 98 at gc.c:9429
    frame #6: 0x0000000100096658 miniruby`gc_writebarrier_incremental(a=4334514520, b=4334514600, objspace=0x0
0000001007d3280) + 61 at gc.c:5963
    frame #7: 0x00000001000968ca miniruby`rb_gc_writebarrier(a=4334514520, b=4334514600) + 127 at gc.c:6009
    frame #8: 0x00000001001eabe0 miniruby`rb_obj_written(a=4334514520, oldv=52, b=4334514600, filename="/Users
/urabe.shyouhei/data/src/pedantic/vm.c", line=821) + 72 at ruby.h:1472
    frame #9: 0x00000001001eac2c miniruby`rb_obj_write(a=4334514520, slot=0x000000010259ff10, b=4334514600, fi
lename="/Users/urabe.shyouhei/data/src/pedantic/vm.c", line=821) + 70 at ruby.h:1489
    frame #10: 0x0000000100208b6f miniruby`vm_proc_create_from_captured(klass=4311027960, captured=0x000000010
2500338, block_type=block_type_ifunc, is_from_method='\0', is_lambda='\x01') + 137 at vm.c:821
    frame #11: 0x0000000100208e5c miniruby`rb_vm_make_proc_lambda(ec=0x00000001007d3548, captured=0x0000000102
500338, klass=4311027960, is_lambda='\x01') + 134 at vm.c:892
```

```
    frame #12: 0x0000000010011f08e miniruby`proc_new(klass=4311027960, is_lambda='\x01') + 445 at proc.c:752
    frame #13: 0x0000000010011f110 miniruby`rb_block_lambda + 27 at proc.c:808
    frame #14: 0x00000001001f24a7 miniruby`call_cfunc_0(func=(miniruby`rb_block_lambda at proc.c:807), recv=43
10991600, argc=0, argv=0x0000000000000000) + 41 at vm_insnhelper.c:1729
    frame #15: 0x00000001002033de miniruby`vm_call0_cfunc_with_frame(ec=0x00000001007d3548, calling=0x00007fff
5fbfb080, ci=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 370 at vm_eval.c:85
    frame #16: 0x00000001002034d9 miniruby`vm_call0_cfunc(ec=0x00000001007d3548, calling=0x00007fff5fbfb080, c
i=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 59 at vm_eval.c:100
    frame #17: 0x0000000010020368f miniruby`vm_call0_body(ec=0x00000001007d3548, calling=0x00007fff5fbfb080, ci
=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 436 at vm_eval.c:131
    frame #18: 0x0000000010020326a miniruby`vm_call0(ec=0x00000001007d3548, recv=4310991600, id=2993, argc=0, a
rgv=0x0000000000000000, me=0x0000000100f48110) + 142 at vm_eval.c:58
    frame #19: 0x0000000100203c60 miniruby`rb_call0(ec=0x00000001007d3548, recv=4310991600, mid=2993, argc=0,
argv=0x0000000000000000, scope=CALL_FCALL, self=4334514640) + 166 at vm_eval.c:296
    frame #20: 0x0000000100204827 miniruby`rb_call(recv=4310991600, mid=2993, argc=0, argv=0x0000000000000000,
 scope=CALL_FCALL) + 84 at vm_eval.c:589
    frame #21: 0x0000000010020518b miniruby`rb_funcallv(recv=4310991600, mid=2993, argc=0, argv=0x0000000000000
000) + 52 at vm_eval.c:815
    frame #22: 0x0000000010012242e miniruby`mlambda(method=0) + 45 at proc.c:2661
    frame #23: 0x0000000100205bac miniruby`rb_iterate0(it_proc=(miniruby`mlambda at proc.c:2660), data1=0, ifu
nc=0x00000001025b71a8, ec=0x00000001007d3548) + 380 at vm_eval.c:1134
    frame #24: 0x0000000100205d16 miniruby`rb_iterate(it_proc=(miniruby`mlambda at proc.c:2660), data1=0, bl_p
roc=(miniruby`bmcall at proc.c:2666), data2=4334514640) + 88 at vm_eval.c:1166
    frame #25: 0x00000001001224c7 miniruby`method_to_proc(method=4334514640) + 43 at proc.c:2701
    frame #26: 0x00000001001f24a7 miniruby`call_cfunc_0(func=(miniruby`method_to_proc at proc.c:2688), recv=43
34514640, argc=0, argv=0x0000000102400568) + 41 at vm_insnhelper.c:1729
    frame #27: 0x00000001001f2f87 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x000000010
2500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 386 at vm_insnhelper.c:19
18
    frame #28: 0x00000001001f30d6 miniruby`vm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500350, ca
lling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 149 at vm_insnhelper.c:1934
    frame #29: 0x00000001001f4319 miniruby`vm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500
350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 239 at vm_insnhelper.c:2232
    frame #30: 0x00000001001f49a4 miniruby`vm_call_method(ec=0x00000001007d3548, cfp=0x0000000102500350, calli
ng=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 117 at vm_insnhelper.c:2355
    frame #31: 0x00000001001f4b7a miniruby`vm_call_general(ec=0x00000001007d3548, reg_cfp=0x0000000102500350,
calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 59 at vm_insnhelper.c:2398
    frame #32: 0x00000001001faf0e miniruby`vm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:
915
    frame #33: 0x000000010020b75d miniruby`vm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771
    frame #34: 0x000000010020c3d1 miniruby`rb_iseq_eval(iseq=0x00000001007f8270) + 52 at vm.c:2008
    frame #35: 0x00000001000caa4a miniruby`rb_load_internal0(ec=0x00000001007d3548, fname=4310799960, wrap=0)
+ 631 at load.c:611
    frame #36: 0x00000001000cab36 miniruby`rb_load_internal(fname=4310799960, wrap=0) + 46 at load.c:642
    frame #37: 0x00000001000cae1d miniruby`rb_f_load(argc=1, argv=0x00000001024004b8) + 217 at load.c:710
    frame #38: 0x00000001001f247c miniruby`call_cfunc_m1(func=(miniruby`rb_f_load at load.c:695), recv=4311327
440, argc=1, argv=0x00000001024004b8) + 47 at vm_insnhelper.c:1723
    frame #39: 0x00000001001f2f87 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x000000010
25003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 386 at vm_insnhelper.c:19
18
    frame #40: 0x00000001001f30d6 miniruby`vm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x00000001025003e0, ca
lling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 149 at vm_insnhelper.c:1934
    frame #41: 0x00000001001f4319 miniruby`vm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500
3e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 239 at vm_insnhelper.c:2232
    frame #42: 0x00000001001f4a2c miniruby`vm_call_method(ec=0x00000001007d3548, cfp=0x00000001025003e0, calli
ng=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 253 at vm_insnhelper.c:2366
    frame #43: 0x00000001001f4b7a miniruby`vm_call_general(ec=0x00000001007d3548, reg_cfp=0x00000001025003e0,
calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 59 at vm_insnhelper.c:2398
    frame #44: 0x00000001001faf0e miniruby`vm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:
915
    frame #45: 0x000000010020b75d miniruby`vm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771
    frame #46: 0x000000010020c40f miniruby`rb_iseq_eval_main(iseq=0x0000000100f21240) + 52 at vm.c:2019
    frame #47: 0x0000000010007c774 miniruby`ruby_exec_internal(n=0x0000000100f21240) + 297 at eval.c:246
    frame #48: 0x0000000010007c89a miniruby`ruby_exec_node(n=0x0000000100f21240) + 36 at eval.c:310
    frame #49: 0x0000000010007c86d miniruby`ruby_run_node(n=0x0000000100f21240) + 62 at eval.c:302
    frame #50: 0x0000000100001399 miniruby`main(argc=9, argv=0x00007fff5fbfdae0) + 113 at main.c:42
    frame #51: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)
```

Revision 62864 - 03/20/2018 02:18 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 61564,61565,61571: [Backport #14270]

```
fix SEGV touching uninitialized memory
```

```
This function can be called from Init_VM().
No assumption can be made about object internals.

(lldb) run
Process 15734 launched: './miniruby' (x86_64)
Process 15734 stopped
* thread #1: tid = 0x1441d4, 0x00000001000bdfcb miniruby`rb_raw_iseq_info(buff="0x0000000100f61f48 [0    ] T_I
MEMO iseq", buff_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273, queue = 'com.apple.main-thread', stop r
eason = EXC_BAD_ACCESS (code=1, address=0x50)
    frame #0: 0x00000001000bdfcb miniruby`rb_raw_iseq_info(buff="0x0000000100f61f48 [0    ] T_IMEMO iseq", buf
f_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273
   9270 static void
   9271 rb_raw_iseq_info(char *buff, const int buff_size, const rb_iseq_t *iseq)
   9272 {
-> 9273     if (iseq->body->location.label) {
   9274         VALUE path = rb_iseq_path(iseq);
   9275         snprintf(buff, buff_size, "%s %s@%s:%d", buff,
   9276                  RSTRING_PTR(iseq->body->location.label),
(lldb) p *iseq
(rb_iseq_t) $0 = {
  flags = 28698
  reserved1 = 0
  body = 0x0000000000000000
  aux = {
    compile_data = 0x0000000000000000
    loader = (obj = 0, index = 0)
    trace_events = 0
  }
}
(lldb) bt
* thread #1: tid = 0x1441d4, 0x00000001000bdfcb miniruby`rb_raw_iseq_info(buff="0x0000000100f61f48 [0    ] T_I
MEMO iseq", buff_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273, queue = 'com.apple.main-thread', stop r
eason = EXC_BAD_ACCESS (code=1, address=0x50)
  * frame #0: 0x00000001000bdfcb miniruby`rb_raw_iseq_info(buff="0x0000000100f61f48 [0    ] T_IMEMO iseq", buf
f_size=256, iseq=0x0000000100f61f48) + 27 at gc.c:9273
    frame #1: 0x00000001000bde72 miniruby`rb_raw_obj_info(buff="0x0000000100f61f48 [0    ] T_IMEMO iseq", buff
_size=256, obj=4311097160) + 2786 at gc.c:9396
    frame #2: 0x00000001000b7c5f miniruby`obj_info(obj=4311097160) + 95 at gc.c:9428
    frame #3: 0x00000001000c16a8 miniruby`newobj_init(klass=0, flags=28698, v1=0, v2=0, v3=0, wb_protected=1,
objspace=0x00000001007ee280, obj=4311097160) + 424 at gc.c:1887
    frame #4: 0x00000001000b44c9 miniruby`newobj_of(klass=0, flags=28698, v1=0, v2=0, v3=0, wb_protected=1) +
217 at gc.c:1970
    frame #5: 0x00000001000b464b miniruby`rb_imemo_new(type=imemo_iseq, v1=0, v2=0, v3=0, v0=0) + 75 at gc.c:2
017
    frame #6: 0x00000001000fd914 miniruby`iseq_imemo_alloc + 36 at iseq.h:156
    frame #7: 0x00000001000f6e1d miniruby`iseq_alloc + 13 at iseq.c:211
    frame #8: 0x00000001000f6bf8 miniruby`rb_iseq_new_with_opt(node=0x0000000000000000, name=4311097200, path=
4311097200, realpath=8, first_lineno=1, parent=0x0000000000000000, type=ISEQ_TYPE_TOP, option=0x0000000100335c
30) + 56 at iseq.c:519
    frame #9: 0x00000001000f6bb6 miniruby`rb_iseq_new(node=0x0000000000000000, name=4311097200, path=431109720
0, realpath=8, parent=0x0000000000000000, type=ISEQ_TYPE_TOP) + 86 at iseq.c:480
    frame #10: 0x0000000100284bb0 miniruby`Init_VM + 1040 at vm.c:3022
    frame #11: 0x00000001000d4f7d miniruby`rb_call_inits + 189 at inits.c:55
    frame #12: 0x0000000010009fe06 miniruby`ruby_setup + 198 at eval.c:61
    frame #13: 0x0000000010009fe5d miniruby`ruby_init + 13 at eval.c:78
    frame #14: 0x00000001000009ed miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 93 at main.c:41
    frame #15: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)

fix SEGV touching uninitialized memory

This function can be called from rb_data_typed_object_zalloc().
No assumption can be made about object internals.

(lldb) run
Process 22135 launched: './miniruby' (x86_64)
Process 22135 stopped
* thread #1: tid = 0x14a3af, 0x0000000010008ac8a miniruby`vm_block_type(block=0x0000000000000000) + 12 at vm_co
re.h:1364, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x18)
    frame #0: 0x0000000010008ac8a miniruby`vm_block_type(block=0x0000000000000000) + 12 at vm_core.h:1364
   1361         break;
   1362     }
   1363 #endif
-> 1364     return block->type;
   1365 }
```

```
   1366
   1367 static inline void
(lldb) bt
* thread #1: tid = 0x14a3af, 0x0000000010008ac8a miniruby`vm_block_type(block=0x0000000000000000) + 12 at vm_co
re.h:1364, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x18)
  * frame #0: 0x0000000010008ac8a miniruby`vm_block_type(block=0x0000000000000000) + 12 at vm_core.h:1364
    frame #1: 0x0000000010008acdb miniruby`vm_block_iseq(block=0x0000000000000000) + 24 at vm_core.h:1399
    frame #2: 0x0000000010008acc1 miniruby`vm_proc_iseq(procval=4310866360) + 32 at vm_core.h:1387
    frame #3: 0x0000000010009cbed miniruby`rb_raw_obj_info(buff="0x0000000100f299b8 [0    ] proc (Proc)", buff_
size=256, obj=4310866360) + 1513 at gc.c:9349
    frame #4: 0x0000000010009cf01 miniruby`obj_info(obj=4310866360) + 98 at gc.c:9428
    frame #5: 0x0000000010008ca1b miniruby`newobj_init(klass=4311027960, flags=12, v1=4298186080, v2=1, v3=0, w
b_protected=32, objspace=0x00000001007cf280, obj=4310866360) + 338 at gc.c:1887
    frame #6: 0x0000000010008cce5 miniruby`newobj_of(klass=4311027960, flags=12, v1=4298186080, v2=1, v3=0, wb_
protected=32) + 171 at gc.c:1970
    frame #7: 0x0000000010008d01d miniruby`rb_data_typed_object_wrap(klass=4311027960, datap=0x0000000000000000
, type=0x0000000100311d60) + 133 at gc.c:2062
    frame #8: 0x0000000010008d04e miniruby`rb_data_typed_object_zalloc(klass=4311027960, size=40, type=0x000000
0100311d60) + 42 at gc.c:2073
    frame #9: 0x0000000010011b459 miniruby`rb_proc_alloc(klass=4311027960) + 36 at proc.c:113
    frame #10: 0x0000000100204d8e miniruby`vm_proc_create_from_captured(klass=4311027960, captured=0x000000010
25003f8, block_type=block_type_iseq, is_from_method='\0', is_lambda='\x01') + 44 at vm.c:814
    frame #11: 0x0000000101002050d8 miniruby`rb_vm_make_proc_lambda(ec=0x00000001007cf548, captured=0x0000000102
5003f8, klass=4311027960, is_lambda='\x01') + 134 at vm.c:892
    frame #12: 0x0000000010011c0d2 miniruby`proc_new(klass=4311027960, is_lambda='\x01') + 445 at proc.c:752
    frame #13: 0x0000000010011c154 miniruby`rb_block_lambda + 27 at proc.c:808
    frame #14: 0x00000001001ee7e3 miniruby`call_cfunc_0(func=(miniruby`rb_block_lambda at proc.c:807), recv=43
10991600, argc=0, argv=0x0000000102400480) + 41 at vm_insnhelper.c:1729
    frame #15: 0x00000001001ef2c3 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007cf548, reg_cfp=0x000000010
25003e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x0000000010253e0f0) + 386 at vm_insnhelper.c:19
18
    frame #16: 0x00000001001ef412 miniruby`vm_call_cfunc(ec=0x00000001007cf548, reg_cfp=0x00000001025003e0, ca
lling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x0000000010253e0f0) + 149 at vm_insnhelper.c:1934
    frame #17: 0x00000001001f0655 miniruby`vm_call_method_each_type(ec=0x00000001007cf548, cfp=0x0000000102500
3e0, calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x0000000010253e0f0) + 239 at vm_insnhelper.c:2232
    frame #18: 0x00000001001f0ce0 miniruby`vm_call_method(ec=0x00000001007cf548, cfp=0x00000001025003e0, calli
ng=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x0000000010253e0f0) + 117 at vm_insnhelper.c:2355
    frame #19: 0x00000001001f0eb6 miniruby`vm_call_general(ec=0x00000001007cf548, reg_cfp=0x00000001025003e0,
calling=0x00007fff5fbfd4d0, ci=0x0000000102537be0, cc=0x0000000010253e0f0) + 59 at vm_insnhelper.c:2398
    frame #20: 0x00000001001f6e61 miniruby`vm_exec_core(ec=0x00000001007cf548, initial=0) + 7480 at insns.def:
850
    frame #21: 0x0000000100207995 miniruby`vm_exec(ec=0x00000001007cf548) + 230 at vm.c:1771
    frame #22: 0x0000000100208647 miniruby`rb_iseq_eval_main(iseq=0x0000000100f29fd0) + 52 at vm.c:2019
    frame #23: 0x0000000010007b750 miniruby`ruby_exec_internal(n=0x0000000100f29fd0) + 297 at eval.c:246
    frame #24: 0x0000000010007b876 miniruby`ruby_exec_node(n=0x0000000100f29fd0) + 36 at eval.c:310
    frame #25: 0x0000000010007b849 miniruby`ruby_run_node(n=0x0000000100f29fd0) + 62 at eval.c:302
    frame #26: 0x0000000100000c05 miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 113 at main.c:42
    frame #27: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)

check an existence of block.

* gc.c (rb_raw_obj_info): check block before using it.

* vm_core.h (vm_block_iseq): r61565 introduced NULL check but this
  check is only needed by `rb_raw_obj_info()` and it is called at GC
  debug mode. Above fix for `rb_raw_obj_info()` solves this problem and
  NULL check should not be needed any more.
```

Revision 62941 - 03/28/2018 05:17 AM - usa (Usaku NAKAMURA)

merge revision(s) 61562,61563,61566,61568,61569: [Backport #14269]

```
fix SEGV touching uninitialized memory

This function can be called from boot_defclass().
No assumption can be made about object internals.

(lldb) run
Process 2386 launched: './miniruby' (x86_64)
Process 2386 stopped
* thread #1: tid = 0x13f3b6, 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variab
le.c:321, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x8)
    frame #0: 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variable.c:321
   318  VALUE
```

```
   319  rb_class_path_cached(VALUE klass)
   320  {
-> 321      st_table *ivtbl = RCLASS_IV_TBL(klass);
   322      st_data_t n;
   323
   324      if (!ivtbl) return Qnil;
(lldb) bt
* thread #1: tid = 0x13f3b6, 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variab
le.c:321, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x8)
  * frame #0: 0x00000001001e0b26 miniruby`rb_class_path_cached(klass=4311373720) + 20 at variable.c:321
    frame #1: 0x000000010009cbd0 miniruby`rb_raw_obj_info(buff="0x0000000100fa5798 [2    ] T_CLASS", buff_size
=256, obj=4311373720) + 1393 at gc.c:9341
    frame #2: 0x000000010009cf16 miniruby`obj_info(obj=4311373720) + 98 at gc.c:9423
    frame #3: 0x000000010008ca87 miniruby`newobj_init(klass=0, flags=66, v1=0, v2=0, v3=0, wb_protected=1, obj
space=0x00000001007cf280, obj=4311373720) + 338 at gc.c:1887
    frame #4: 0x000000010008cd51 miniruby`newobj_of(klass=0, flags=66, v1=0, v2=0, v3=0, wb_protected=1) + 171
 at gc.c:1970
    frame #5: 0x000000010008ce1b miniruby`rb_wb_protected_newobj_of(klass=0, flags=66) + 54 at gc.c:1990
    frame #6: 0x0000000100027563 miniruby`class_alloc(flags=2, klass=0) + 46 at class.c:165
    frame #7: 0x000000010002761a miniruby`rb_class_boot(super=0) + 35 at class.c:203
    frame #8: 0x0000000100028612 miniruby`boot_defclass(name="BasicObject", super=0) + 28 at class.c:537
    frame #9: 0x000000010002868b miniruby`Init_class_hierarchy + 26 at class.c:548
    frame #10: 0x00000001000efe69 miniruby`InitVM_Object + 9 at object.c:3892
    frame #11: 0x00000001000f138e miniruby`Init_Object + 57 at object.c:4122
    frame #12: 0x00000001000a59bd miniruby`rb_call_inits + 29 at inits.c:23
    frame #13: 0x000000010007af30 miniruby`ruby_setup + 229 at eval.c:61
    frame #14: 0x000000010007af7e miniruby`ruby_init + 13 at eval.c:78
    frame #15: 0x0000000100000c58 miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 88 at main.c:41
    frame #16: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)

fix SEGV touching uninitialized memory

This function can be called from InitVM_Object().
No assumption can be made about object internals.

(lldb) run
Process 10675 launched: './miniruby' (x86_64)
Process 10675 stopped
* thread #1: tid = 0x14252c, 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0    ] T_IM
EMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383, queue = 'com.apple.main-thread', stop reason =
EXC_BAD_ACCESS (code=1, address=0x0)
    frame #0: 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0    ] T_IMEMO ment", buff
_size=256, obj=4311487880) + 2489 at gc.c:9383
   9380                const rb_method_entry_t *me = &RANY(obj)->as.imemo.ment;
   9381                snprintf(buff, buff_size, "%s (called_id: %s, type: %s, alias: %d, owner: %s, defined_
class: %s)", buff,
   9382                         rb_id2name(me->called_id),
-> 9383                         method_type_name(me->def->type),
   9384                         me->def->alias_count,
   9385                         obj_info(me->owner),
   9386                         obj_info(me->defined_class));
(lldb) p *me
(rb_method_entry_t) $0 = {
  flags = 24602
  defined_class = 4311488400
  def = 0x0000000000000000
  called_id = 3057
  owner = 4311488400
}
(lldb) bt
* thread #1: tid = 0x14252c, 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0    ] T_IM
EMO ment", buff_size=256, obj=4311487880) + 2489 at gc.c:9383, queue = 'com.apple.main-thread', stop reason =
EXC_BAD_ACCESS (code=1, address=0x0)
  * frame #0: 0x00000001000bdda9 miniruby`rb_raw_obj_info(buff="0x0000000100fc1588 [0    ] T_IMEMO ment", buff
_size=256, obj=4311487880) + 2489 at gc.c:9383
    frame #1: 0x00000001000b7cbf miniruby`obj_info(obj=4311487880) + 95 at gc.c:9423
    frame #2: 0x00000001000c16a8 miniruby`newobj_init(klass=4311488400, flags=24602, v1=0, v2=3057, v3=4311488
400, wb_protected=1, objspace=0x00000001007ee280, obj=4311487880) + 424 at gc.c:1887
    frame #3: 0x00000001000b4529 miniruby`newobj_of(klass=4311488400, flags=24602, v1=0, v2=3057, v3=431148840
0, wb_protected=1) + 217 at gc.c:1970
    frame #4: 0x00000001000b46ab miniruby`rb_imemo_new(type=imemo_ment, v1=0, v2=3057, v3=4311488400, v0=43114
88400) + 75 at gc.c:2017
    frame #5: 0x00000001002773b4 miniruby`rb_method_entry_alloc(called_id=3057, owner=4311488400, defined_clas
s=4311488400, def=0x0000000000000000) + 52 at vm_method.c:368
```

```
    frame #6: 0x0000000100277307 miniruby`rb_method_entry_create(called_id=3057, klass=4311488400, visi=METHOD
_VISI_PRIVATE, def=0x0000000000000000) + 71 at vm_method.c:389
    frame #7: 0x00000001002784c7 miniruby`rb_method_entry_make(klass=4311488400, mid=3057, defined_class=43114
88400, visi=METHOD_VISI_PRIVATE, type=VM_METHOD_TYPE_CFUNC, def=0x0000000000000000, original_id=3057, opts=0x0
0007fff5fbfd9e8) + 1207 at vm_method.c:594
    frame #8: 0x00000001002770f9 miniruby`rb_add_method(klass=4311488400, mid=3057, type=VM_METHOD_TYPE_CFUNC,
 opts=0x00007fff5fbfd9e8, visi=METHOD_VISI_PRIVATE) + 73 at vm_method.c:650
    frame #9: 0x000000010027708a miniruby`rb_add_method_cfunc(klass=4311488400, mid=3057, func=(miniruby`rb_ob
j_dummy at object.c:1125), argc=0, visi=METHOD_VISI_PRIVATE) + 138 at vm_method.c:137
    frame #10: 0x00000001000391e4 miniruby`rb_define_private_method(klass=4311488400, name="initialize", func=
(miniruby`rb_obj_dummy at object.c:1125), argc=0) + 68 at class.c:1529
    frame #11: 0x0000000010013f5bf miniruby`InitVM_Object + 47 at object.c:3905
    frame #12: 0x0000000100142ffd miniruby`Init_Object + 61 at object.c:4122
    frame #13: 0x00000001000d4edd miniruby`rb_call_inits + 29 at inits.c:23
    frame #14: 0x0000000010009fe66 miniruby`ruby_setup + 198 at eval.c:61
    frame #15: 0x0000000010009febd miniruby`ruby_init + 13 at eval.c:78
    frame #16: 0x0000000100000a4d miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 93 at main.c:41
    frame #17: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)

fix SEGV touching uninitialized local variable

This imemo_name is used uninitialized because the switch
above does not cover all possible imemo types.

(lldb) run
Process 26068 launched: './miniruby' (x86_64)
Process 26068 stopped
* thread #1: tid = 0x14ba96, 0x00007fff8a402132 libsystem_c.dylib`strlen + 18, queue = 'com.apple.main-thread'
, stop reason = EXC_BAD_ACCESS (code=1, address=0xfffffffffffffff0)
    frame #0: 0x00007fff8a402132 libsystem_c.dylib`strlen + 18
libsystem_c.dylib`strlen:
->  0x7fff8a402132 <+18>: pcmpeqb (%rdi), %xmm0
    0x7fff8a402136 <+22>: pmovmskb %xmm0, %esi
    0x7fff8a40213a <+26>: andq   $0xf, %rcx
    0x7fff8a40213e <+30>: orq    $-0x1, %rax
(lldb) bt
* thread #1: tid = 0x14ba96, 0x00007fff8a402132 libsystem_c.dylib`strlen + 18, queue = 'com.apple.main-thread'
, stop reason = EXC_BAD_ACCESS (code=1, address=0xfffffffffffffff0)
  * frame #0: 0x00007fff8a402132 libsystem_c.dylib`strlen + 18
    frame #1: 0x00000001001f1531 miniruby`BSD_vfprintf(fp=0x00007fff5fbfc9e0, fmt0="%s %s", ap=0x00007fff5fbfc
bf0) + 5873 at vsnprintf.c:1026
    frame #2: 0x00000001001ef213 miniruby`ruby_do_vsnprintf(str="0x0000000100f46450 [0    ] T_IMEMO", n=256, f
mt="%s %s", ap=0x00007fff5fbfcbf0) + 131 at sprintf.c:1285
    frame #3: 0x00000001001ef3ea miniruby`ruby_snprintf(str="0x0000000100f46450 [0    ] T_IMEMO", n=256, fmt="
%s %s") + 426 at sprintf.c:1300
    frame #4: 0x00000001000bdc61 miniruby`rb_raw_obj_info(buff="0x0000000100f46450 [0    ] T_IMEMO", buff_size
=256, obj=4310983760) + 2353 at gc.c:9376
    frame #5: 0x00000001000b7bff miniruby`obj_info(obj=4310983760) + 95 at gc.c:9428
    frame #6: 0x00000001000c1658 miniruby`newobj_init(klass=0, flags=36890, v1=0, v2=4303040512, v3=4310983800
, wb_protected=1, objspace=0x00000001007ee280, obj=4310983760) + 424 at gc.c:1887
    frame #7: 0x00000001000b4469 miniruby`newobj_of(klass=0, flags=36890, v1=0, v2=4303040512, v3=4310983800,
 wb_protected=1) + 217 at gc.c:1970
    frame #8: 0x00000001000b45eb miniruby`rb_imemo_new(type=imemo_ast, v1=0, v2=4303040512, v3=4310983800, v0=
0) + 75 at gc.c:2017
    frame #9: 0x000000010011daed miniruby`rb_ast_new + 61 at node.c:1146
    frame #10: 0x0000000100160e15 miniruby`rb_parser_compile_file_path(vparser=4310984400, fname=4310984960, f
ile=4310984080, start=1) + 53 at parse.y:5776
    frame #11: 0x00000001001e18ea miniruby`load_file_internal(argp_v=140734799795024) + 1834 at ruby.c:1907
    frame #12: 0x00000001000a1bb5 miniruby`rb_ensure(b_proc=(miniruby`load_file_internal at ruby.c:1795), data
1=140734799795024, e_proc=(miniruby`restore_load_file at ruby.c:2007), data2=140734799795024) + 245 at eval.c:
1037
    frame #13: 0x00000001001df4a4 miniruby`load_file(parser=4310984400, fname=4310984960, f=4310984080, script
=1, opt=0x00007fff5fbfda28) + 100 at ruby.c:2026
    frame #14: 0x00000001001e084e miniruby`process_options(argc=0, argv=0x00007fff5fbfdc00, opt=0x00007fff5fbf
da28) + 3454 at ruby.c:1682
    frame #15: 0x00000001001dfaae miniruby`ruby_process_options(argc=2, argv=0x00007fff5fbfdbf0) + 238 at ruby
.c:2257
    frame #16: 0x0000000010009ff43 miniruby`ruby_options(argc=2, argv=0x00007fff5fbfdbf0) + 211 at eval.c:105
    frame #17: 0x0000000100000989 miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 105 at main.c:42
    frame #18: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb) up 4
frame #4: 0x00000001000bdc61 miniruby`rb_raw_obj_info(buff="0x0000000100f46450 [0    ] T_IMEMO", buff_size=256
, obj=4310983760) + 2353 at gc.c:9376
   9373 #undef IMEMO_NAME
```

```
   9374               default: UNREACHABLE;
   9375             }
-> 9376             snprintf(buff, buff_size, "%s %s", buff, imemo_name);
   9377
   9378             switch (imemo_type(obj)) {
   9379               case imemo_ment: {
(lldb) p imemo_name
(const char *) $0 = 0xffffffffffffffff
(lldb) p imemo_type(obj)
(imemo_type) $1 = imemo_ast
(lldb)
```

fix SEGV inspecting already freed objects

obj_info() assumes the given object is alive. Passing freed
objects to it results in SEGV.

```
(lldb) run
Process 29718 launched: './miniruby' (x86_64)
Process 29718 stopped
* thread #1: tid = 0x3082c5, 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:26
9, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
    frame #0: 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:269
   266      }
   267      else {
   268          VM_ASSERT(RB_TYPE_P(pathobj, T_ARRAY));
-> 269          return RARRAY_AREF(pathobj, PATHOBJ_PATH);
   270      }
   271  }
   272
(lldb) bt
* thread #1: tid = 0x3082c5, 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:26
9, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
  * frame #0: 0x00000001000bfaab miniruby`pathobj_path(pathobj=4478683640) + 70 at vm_core.h:269
    frame #1: 0x00000001000c25ff miniruby`rb_iseq_path(iseq=0x000000010af34a20) + 32 at iseq.c:723
    frame #2: 0x000000010009db09 miniruby`rb_raw_iseq_info(buff="0x000000010af34a20 [1    ] T_IMEMO iseq", buf
f_size=256, iseq=0x000000010af34a20) + 69 at gc.c:9274
    frame #3: 0x000000010009e45a miniruby`rb_raw_obj_info(buff="0x000000010af34a20 [1    ] T_IMEMO iseq", buff
_size=256, obj=4478683680) + 2191 at gc.c:9397
    frame #4: 0x000000010009e4d5 miniruby`obj_info(obj=4478683680) + 98 at gc.c:9429
    frame #5: 0x0000000100091ae3 miniruby`gc_page_sweep(objspace=0x00000001007d3280, heap=0x00000001007d32a0,
 sweep_page=0x000000010ae07bc0) + 622 at gc.c:3529
    frame #6: 0x000000010009206a miniruby`gc_sweep_step(objspace=0x00000001007d3280, heap=0x00000001007d32a0)
 + 188 at gc.c:3705
    frame #7: 0x0000000100092254 miniruby`gc_sweep_continue(objspace=0x00000001007d3280, heap=0x00000001007d32
a0) + 133 at gc.c:3772
    frame #8: 0x000000010008d7f9 miniruby`heap_prepare(objspace=0x00000001007d3280, heap=0x00000001007d32a0) +
 48 at gc.c:1746
    frame #9: 0x000000010008d8a1 miniruby`heap_get_freeobj_from_next_freepage(objspace=0x00000001007d3280, hea
p=0x00000001007d32a0) + 37 at gc.c:1769
    frame #10: 0x000000010008d98d miniruby`heap_get_freeobj(objspace=0x00000001007d3280, heap=0x00000001007d32
a0) + 83 at gc.c:1803
    frame #11: 0x000000010008dcb0 miniruby`newobj_slowpath(klass=4334386280, flags=5, v1=0, v2=0, v3=0, objspa
ce=0x00000001007d3280, wb_protected=1) + 220 at gc.c:1930
    frame #12: 0x000000010008dd6c miniruby`newobj_slowpath_wb_protected(klass=4334386280, flags=5, v1=0, v2=0,
 v3=0, objspace=0x00000001007d3280) + 76 at gc.c:1942
    frame #13: 0x000000010008dea1 miniruby`newobj_of(klass=4334386280, flags=5, v1=0, v2=0, v3=0, wb_protected
=1) + 221 at gc.c:1974
    frame #14: 0x000000010008df39 miniruby`rb_wb_protected_newobj_of(klass=4334386280, flags=5) + 54 at gc.c:1
990
    frame #15: 0x0000000100195f7c miniruby`str_alloc(klass=4334386280) + 29 at string.c:692
    frame #16: 0x0000000100195fe9 miniruby`str_new0(klass=4334386280, ptr="gitm", len=4, termlen=1) + 73 at st
ring.c:714
    frame #17: 0x000000010019633e miniruby`rb_enc_str_new(ptr="gitm", len=4, enc=0x00000001025d50a0) + 81 at s
tring.c:766
    frame #18: 0x000000010010a80a miniruby`parser_str_new(p="gitm", n=4, enc=0x00000001025d50a0, func=66, enc0
=0x00000001025d50a0) + 50 at parse.y:5817
    frame #19: 0x000000010010ce1a miniruby`parser_parse_string(parser=0x00000001042ac5c0, quote=0x000000010460
c028) + 795 at parse.y:6675
    frame #20: 0x00000001001120bd miniruby`parser_yylex(parser=0x00000001042ac5c0) + 159 at parse.y:8281
    frame #21: 0x0000000100115068 miniruby`yylex(lval=0x00007fff5fbf9948, yylloc=0x00007fff5fbf9ab0, parser=0x
00000001042ac5c0) + 55 at parse.y:8931
    frame #22: 0x00000001000fc79f miniruby`ruby_yyparse(parser=0x00000001042ac5c0) + 1198 at parse.c:5798
    frame #23: 0x0000000100109f5a miniruby`yycompile0(arg=4364879296) + 317 at parse.y:5595
    frame #24: 0x0000000100214ef0 miniruby`rb_suppress_tracing(func=(miniruby`yycompile0 at parse.y:5565), arg
```

```
=4364879296) + 349 at vm_trace.c:397
    frame #25: 0x0000000010010a1df miniruby`yycompile(parser=0x00000001042ac5c0, fname=4443743440, line=1) + 12
6 at parse.y:5637
    frame #26: 0x0000000010010a4c1 miniruby`parser_compile_string(vparser=4443743480, fname=4443743440, s=44437
43520, line=1) + 191 at parse.y:5706
    frame #27: 0x0000000010010a5b7 miniruby`rb_parser_compile_string_path(vparser=4443743480, f=4443743440, s=4
443743520, line=1) + 58 at parse.y:5730
    frame #28: 0x0000000100206025 miniruby`eval_make_iseq(src=4443743520, fname=4443743440, line=1, bind=0x000
0000000000000, base_block=0x00007fff5fbfb370) + 266 at vm_eval.c:1274
    frame #29: 0x0000000100206153 miniruby`eval_string_with_cref(self=4334412520, src=4443743520, cref=0x00000
00000000000, file=52, line=1) + 197 at vm_eval.c:1307
    frame #30: 0x0000000100206389 miniruby`rb_f_eval(argc=1, argv=0x0000000102400eb8, self=4334412520) + 219 a
t vm_eval.c:1382
    frame #31: 0x00000001001f247c miniruby`call_cfunc_m1(func=(miniruby`rb_f_eval at vm_eval.c:1364), recv=433
4412520, argc=1, argv=0x0000000102400eb8) + 47 at vm_insnhelper.c:1723
    frame #32: 0x00000001001f2f87 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x000000010
2500d80, calling=0x00007fff5fbfbf50, ci=0x000000010263f240, cc=0x0000000100749b50) + 386 at vm_insnhelper.c:19
18
    frame #33: 0x00000001001f30d6 miniruby`vm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500d80, ca
lling=0x00007fff5fbfbf50, ci=0x000000010263f240, cc=0x0000000100749b50) + 149 at vm_insnhelper.c:1934
    frame #34: 0x00000001001faf0e miniruby`vm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:
915
    frame #35: 0x0000000010020b75d miniruby`vm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771
    frame #36: 0x0000000102093f8 miniruby`invoke_block(ec=0x00000001007d3548, iseq=0x000000010252d7f0, self=4
334412520, captured=0x0000000102500df8, cref=0x0000000000000000, type=572653569, opt_pc=0) + 224 at vm.c:988
    frame #37: 0x0000000100209766 miniruby`invoke_iseq_block_from_c(ec=0x00000001007d3548, captured=0x00000001
02500df8, self=4334412520, argc=0, argv=0x0000000000000000, passed_block_handler=0, cref=0x0000000000000000, i
s_lambda=0) + 389 at vm.c:1040
    frame #38: 0x0000000100209824 miniruby`invoke_block_from_c_bh(ec=0x00000001007d3548, block_handler=4333768
185, argc=0, argv=0x0000000000000000, passed_block_handler=0, cref=0x0000000000000000, is_lambda=0, force_bloc
karg=0) + 138 at vm.c:1058
    frame #39: 0x00000001002099d0 miniruby`vm_yield(ec=0x00000001007d3548, argc=0, argv=0x0000000000000000) +
69 at vm.c:1103
    frame #40: 0x0000000100205623 miniruby`rb_yield_0(argc=0, argv=0x0000000000000000) + 40 at vm_eval.c:970
    frame #41: 0x0000000100205964 miniruby`loop_i + 19 at vm_eval.c:1049
    frame #42: 0x0000000010007db07 miniruby`rb_rescue2(b_proc=(miniruby`loop_i at vm_eval.c:1047), data1=0, r_p
roc=(miniruby`loop_stop at vm_eval.c:1056), data2=0) + 369 at eval.c:896
    frame #43: 0x0000000100205a2e miniruby`rb_f_loop(self=4334412520) + 121 at vm_eval.c:1100
    frame #44: 0x00000001001f24a7 miniruby`call_cfunc_0(func=(miniruby`rb_f_loop at vm_eval.c:1098), recv=4334
412520, argc=0, argv=0x0000000102400e80) + 41 at vm_insnhelper.c:1729
    frame #45: 0x00000001001f2f87 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x000000010
2500de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 386 at vm_insnhelper.c:19
18
    frame #46: 0x00000001001f30d6 miniruby`vm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500de0, ca
lling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 149 at vm_insnhelper.c:1934
    frame #47: 0x00000001001f4319 miniruby`vm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500
de0, calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 239 at vm_insnhelper.c:2232
    frame #48: 0x00000001001f4a2c miniruby`vm_call_method(ec=0x00000001007d3548, cfp=0x0000000102500de0, calli
ng=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 253 at vm_insnhelper.c:2366
    frame #49: 0x00000001001f4b7a miniruby`vm_call_general(ec=0x00000001007d3548, reg_cfp=0x0000000102500de0,
calling=0x00007fff5fbfd4d0, ci=0x000000010263bbf0, cc=0x0000000102642118) + 59 at vm_insnhelper.c:2398
    frame #50: 0x00000001001fab2f miniruby`vm_exec_core(ec=0x00000001007d3548, initial=0) + 7480 at insns.def:
850
    frame #51: 0x0000000010020b75d miniruby`vm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771
    frame #52: 0x0000000010020c40f miniruby`rb_iseq_eval_main(iseq=0x000000010252dd90) + 52 at vm.c:2019
    frame #53: 0x0000000010007c768 miniruby`ruby_exec_internal(n=0x000000010252dd90) + 297 at eval.c:246
    frame #54: 0x0000000010007c88e miniruby`ruby_exec_node(n=0x000000010252dd90) + 36 at eval.c:310
    frame #55: 0x0000000010007c861 miniruby`ruby_run_node(n=0x000000010252dd90) + 62 at eval.c:302
    frame #56: 0x000000010000138d miniruby`main(argc=2, argv=0x00007fff5fbfdbf0) + 113 at main.c:42
    frame #57: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb) p ((struct RVALUE*)pathobj)->as.basic
(RBasic) $0 = (flags = 0, klass = 4478683600)
(lldb)

fix SEGV inspecting uninitialized objects

obj_info() assumes the given object is alive.  OTOH
gc_writebarrier_incremental is called before or in middle of
object initialization.  Can casue SEGV.

(lldb) run
Process 48188 launched: './miniruby' (x86_64)
Process 48188 stopped
* thread #1: tid = 0x30fd53, 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=52512912222254831145) + 12 at rub
y.h:2072, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=EXC_I386_GPFLT)
```

```
    frame #0: 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072
   2069 static inline const VALUE *
   2070 rb_array_const_ptr(VALUE a)
   2071 {
-> 2072     return FIX_CONST_VALUE_PTR((RBASIC(a)->flags & RARRAY_EMBED_FLAG) ?
   2073         RARRAY(a)->as.ary : RARRAY(a)->as.heap.ptr);
   2074 }
   2075
(lldb) bt
* thread #1: tid = 0x30fd53, 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at rub
y.h:2072, queue = 'com.apple.main-thread', stop reason = EXC_BAD_ACCESS (code=EXC_I386_GPFLT)
  * frame #0: 0x00000001000bf7a9 miniruby`rb_array_const_ptr(a=5251291222225483145) + 12 at ruby.h:2072
    frame #1: 0x00000001000bfaab miniruby`pathobj_path(pathobj=5251291222225483145) + 70 at vm_core.h:269
    frame #2: 0x00000001000c25ff miniruby`rb_iseq_path(iseq=0x00000001025b71a8) + 32 at iseq.c:723
    frame #3: 0x0000000010009db09 miniruby`rb_raw_iseq_info(buff="0x00000001025b7158 [0    ] proc (Proc)", buff
_size=256, iseq=0x00000001025b71a8) + 69 at gc.c:9274
    frame #4: 0x0000000010009e1d5 miniruby`rb_raw_obj_info(buff="0x00000001025b7158 [0    ] proc (Proc)", buff_
size=256, obj=4334514520) + 1546 at gc.c:9351
    frame #5: 0x0000000010009e4d5 miniruby`obj_info(obj=4334514520) + 98 at gc.c:9429
    frame #6: 0x0000000100096658 miniruby`gc_writebarrier_incremental(a=4334514520, b=4334514600, objspace=0x0
0000001007d3280) + 61 at gc.c:5963
    frame #7: 0x00000001000968ca miniruby`rb_gc_writebarrier(a=4334514520, b=4334514600) + 127 at gc.c:6009
    frame #8: 0x00000001001eabe0 miniruby`rb_obj_written(a=4334514520, oldv=52, b=4334514600, filename="/Users
/urabe.shyouhei/data/src/pedantic/vm.c", line=821) + 72 at ruby.h:1472
    frame #9: 0x00000001001eac2c miniruby`rb_obj_write(a=4334514520, slot=0x0000000010259ff10, b=4334514600, fi
lename="/Users/urabe.shyouhei/data/src/pedantic/vm.c", line=821) + 70 at ruby.h:1489
    frame #10: 0x0000000100208b6f miniruby`vm_proc_create_from_captured(klass=4311027960, captured=0x000000010
2500338, block_type=block_type_ifunc, is_from_method='\0', is_lambda='\x01') + 137 at vm.c:821
    frame #11: 0x0000000100208e5c miniruby`rb_vm_make_proc_lambda(ec=0x00000001007d3548, captured=0x0000000102
500338, klass=4311027960, is_lambda='\x01') + 134 at vm.c:892
    frame #12: 0x0000000010011f08e miniruby`proc_new(klass=4311027960, is_lambda='\x01') + 445 at proc.c:752
    frame #13: 0x0000000010011f110 miniruby`rb_block_lambda + 27 at proc.c:808
    frame #14: 0x00000001001f24a7 miniruby`call_cfunc_0(func=(miniruby`rb_block_lambda at proc.c:807), recv=43
10991600, argc=0, argv=0x0000000000000000) + 41 at vm_insnhelper.c:1729
    frame #15: 0x00000001002033de miniruby`vm_call0_cfunc_with_frame(ec=0x00000001007d3548, calling=0x00007fff
5fbfb080, ci=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 370 at vm_eval.c:85
    frame #16: 0x00000001002034d9 miniruby`vm_call0_cfunc(ec=0x00000001007d3548, calling=0x00007fff5fbfb080, c
i=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 59 at vm_eval.c:100
    frame #17: 0x0000000010020368f miniruby`vm_call0_body(ec=0x00000001007d3548, calling=0x00007fff5fbfb080, ci
=0x00007fff5fbfb070, cc=0x00007fff5fbfb0a0, argv=0x0000000000000000) + 436 at vm_eval.c:131
    frame #18: 0x0000000010020326a miniruby`vm_call0(ec=0x00000001007d3548, recv=4310991600, id=2993, argc=0, a
rgv=0x0000000000000000, me=0x0000000100f48110) + 142 at vm_eval.c:58
    frame #19: 0x0000000100203c60 miniruby`rb_call0(ec=0x00000001007d3548, recv=4310991600, mid=2993, argc=0,
 argv=0x0000000000000000, scope=CALL_FCALL, self=4334514640) + 166 at vm_eval.c:296
    frame #20: 0x0000000100204827 miniruby`rb_call(recv=4310991600, mid=2993, argc=0, argv=0x0000000000000000,
  scope=CALL_FCALL) + 84 at vm_eval.c:589
    frame #21: 0x000000010020518b miniruby`rb_funcallv(recv=4310991600, mid=2993, argc=0, argv=0x0000000000000
000) + 52 at vm_eval.c:815
    frame #22: 0x0000000010012242e miniruby`mlambda(method=0) + 45 at proc.c:2661
    frame #23: 0x0000000100205bac miniruby`rb_iterate0(it_proc=(miniruby`mlambda at proc.c:2660), data1=0, ifu
nc=0x00000001025b71a8, ec=0x00000001007d3548) + 380 at vm_eval.c:1134
    frame #24: 0x0000000100205d16 miniruby`rb_iterate(it_proc=(miniruby`mlambda at proc.c:2660), data1=0, bl_p
roc=(miniruby`bmcall at proc.c:2666), data2=4334514640) + 88 at vm_eval.c:1166
    frame #25: 0x00000001001224c7 miniruby`method_to_proc(method=4334514640) + 43 at proc.c:2701
    frame #26: 0x00000001001f24a7 miniruby`call_cfunc_0(func=(miniruby`method_to_proc at proc.c:2688), recv=43
34514640, argc=0, argv=0x0000000102400568) + 41 at vm_insnhelper.c:1729
    frame #27: 0x00000001001f2f87 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x000000010
2500350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 386 at vm_insnhelper.c:19
18
    frame #28: 0x00000001001f30d6 miniruby`vm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x0000000102500350, ca
lling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 149 at vm_insnhelper.c:1934
    frame #29: 0x00000001001f4319 miniruby`vm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500
350, calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 239 at vm_insnhelper.c:2232
    frame #30: 0x00000001001f49a4 miniruby`vm_call_method(ec=0x00000001007d3548, cfp=0x0000000102500350, calli
ng=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 117 at vm_insnhelper.c:2355
    frame #31: 0x00000001001f4b7a miniruby`vm_call_general(ec=0x00000001007d3548, reg_cfp=0x0000000102500350,
 calling=0x00007fff5fbfc030, ci=0x0000000100f2ec70, cc=0x0000000102735718) + 59 at vm_insnhelper.c:2398
    frame #32: 0x00000001001faf0e miniruby`vm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:
915
    frame #33: 0x0000000010020b75d miniruby`vm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771
    frame #34: 0x0000000010020c3d1 miniruby`rb_iseq_eval(iseq=0x00000001007f8270) + 52 at vm.c:2008
    frame #35: 0x00000001000caa4a miniruby`rb_load_internal0(ec=0x00000001007d3548, fname=4310799960, wrap=0)
+ 631 at load.c:611
    frame #36: 0x00000001000cab36 miniruby`rb_load_internal(fname=4310799960, wrap=0) + 46 at load.c:642
    frame #37: 0x00000001000cae1d miniruby`rb_f_load(argc=1, argv=0x00000001024004b8) + 217 at load.c:710
```

```
    frame #38: 0x00000001001f247c miniruby`call_cfunc_m1(func=(miniruby`rb_f_load at load.c:695), recv=4311327
440, argc=1, argv=0x00000001024004b8) + 47 at vm_insnhelper.c:1723
    frame #39: 0x00000001001f2f87 miniruby`vm_call_cfunc_with_frame(ec=0x00000001007d3548, reg_cfp=0x000000010
25003e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 386 at vm_insnhelper.c:19
18
    frame #40: 0x00000001001f30d6 miniruby`vm_call_cfunc(ec=0x00000001007d3548, reg_cfp=0x00000001025003e0, ca
lling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 149 at vm_insnhelper.c:1934
    frame #41: 0x00000001001f4319 miniruby`vm_call_method_each_type(ec=0x00000001007d3548, cfp=0x0000000102500
3e0, calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 239 at vm_insnhelper.c:2232
    frame #42: 0x00000001001f4a2c miniruby`vm_call_method(ec=0x00000001007d3548, cfp=0x00000001025003e0, calli
ng=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 253 at vm_insnhelper.c:2366
    frame #43: 0x00000001001f4b7a miniruby`vm_call_general(ec=0x00000001007d3548, reg_cfp=0x00000001025003e0,
calling=0x00007fff5fbfd3e0, ci=0x0000000102541070, cc=0x0000000100f9e918) + 59 at vm_insnhelper.c:2398
    frame #44: 0x00000001001faf0e miniruby`vm_exec_core(ec=0x00000001007d3548, initial=0) + 8471 at insns.def:
915
    frame #45: 0x000000010020b75d miniruby`vm_exec(ec=0x00000001007d3548) + 230 at vm.c:1771
    frame #46: 0x000000010020c40f miniruby`rb_iseq_eval_main(iseq=0x0000000100f21240) + 52 at vm.c:2019
    frame #47: 0x000000010007c774 miniruby`ruby_exec_internal(n=0x0000000100f21240) + 297 at eval.c:246
    frame #48: 0x000000010007c89a miniruby`ruby_exec_node(n=0x0000000100f21240) + 36 at eval.c:310
    frame #49: 0x000000010007c86d miniruby`ruby_run_node(n=0x0000000100f21240) + 62 at eval.c:302
    frame #50: 0x0000000100001399 miniruby`main(argc=9, argv=0x00007fff5fbfdae0) + 113 at main.c:42
    frame #51: 0x00007fff88eda5ad libdyld.dylib`start + 1
(lldb)
```

Revision 65148 - 10/18/2018 05:51 AM - shyouhei (Shyouhei Urabe)

fix SEGV in rb_raw_obj_info()

This funcion can be called from inside of rb_ast_new().
Should add appropriate case branches.

(lldb) run
Process 9135 launched: './miniruby' (x86_64)
Process 9135 stopped

- thread #1: tid = 0xdf36b, 0x00000001000ca4f9 minirubyrb_raw_obj_info(buff="0x000000010205d158 [0] T_IMEMO", buff_size=256, obj=4328903000) + 2361 at gc.c:9617, queue = 'com.apple.main-thread', stop reason = EXC_BAD_INSTRUCTION (code=EXC_I386_INVOP, subcode=0x0) frame #0: 0x00000001000ca4f9 minirubyrb_raw_obj_info(buff="0x000000010205d158 [0] T_IMEMO", buff_size=256, obj=4328903000) + 2361 at gc.c:9617 9614 IMEMO_NAME(iseq); 9615 IMEMO_NAME(tmpbuf); 9616 #undef IMEMO_NAME -> 9617 default: UNREACHABLE; 9618 } 9619 snprintf(buff, buff_size, "%s %s", buff, imemo_name); 9620 (lldb) bt
- thread #1: tid = 0xdf36b, 0x00000001000ca4f9 miniruby`rb_raw_obj_info(buff="0x000000010205d158 [0] T_IMEMO", buff_size=256, obj=4328903000) + 2361 at gc.c:9617, queue = 'com.apple.main-thread', stop reason = EXC_BAD_INSTRUCTION (code=EXC_I386_INVOP, subcode=0x0)
  - frame #0: 0x00000001000ca4f9 minirubyrb_raw_obj_info(buff="0x000000010205d158 [0] T_IMEMO", buff_size=256, obj=4328903000) + 2361 at gc.c:9617 frame #1: 0x00000001000c433f minirubyobj_info(obj=4328903000) + 95 at gc.c:9671 frame #2: 0x00000001000ce2ac minirubynewobj_init(klass=4302478608, flags=36890, v1=0, v2=0, v3=0, wb_protected=1, objspace=0x0000000101800410, obj=4328903000) + 444 at gc.c:1882 frame #3: 0x00000001000c0a49 minirubynewobj_of(klass=4302478608, flags=36890, v1=0, v2=0, v3=0, wb_protected=1) + 217 at gc.c:1968 frame #4: 0x00000001000c0bcb minirubyrb_imemo_new(type=imemo_ast, v1=0, v2=0, v3=0, v0=4302478608) + 75 at gc.c:2017 frame #5: 0x0000000100148f2a minirubyrb_ast_new + 58 at node.c:1118 frame #6: 0x000000010018d9e2 minirubyyycompile(vparser=4328903720, p=0x0000000100729670, fname=4328903160, line=1) + 98 at parse.y:4925 frame #7: 0x000000010018d66f minirubyparser_compile_string(vparser=4328903720, fname=4328903160, s=4328904440, line=1) + 143 at parse.y:4995 frame #8: 0x000000010018d768 minirubyrb_parser_compile_string_path(vparser=4328903720, f=4328903160, s=4328904440, line=1) + 56 at parse.y:5015 frame #9: 0x000000010018d71e minirubyrb_parser_compile_string(vparser=4328903720, f="-e", s=4328904440, line=1) + 62 at parse.y:5008 frame #10: 0x0000001002130d5 minirubyprocess_options(argc=0, argv=0x00007fff5fbfdc08, opt=0x00007fff5fbfd9e8) + 3477 at ruby.c:1754 frame #11: 0x00000001002122dd minirubyruby_process_options(argc=2, argv=0x00007fff5fbfdbf8) + 285 at ruby.c:2332 frame #12: 0x00000001000aa966 minirubyruby_options(argc=2, argv=0x00007fff5fbfdbf8) + 262 at eval.c:118 frame #13: 0x0000000100000ed4 minirubymain(argc=2, argv=0x00007fff5fbfdbf8) + 116 at main.c:42 frame #14: 0x00007fff933845ad libdyld.dylibstart + 1 frame #15: 0x00007fff933845ad libdyld.dylibstart + 1 (lldb)

Revision 65148 - 10/18/2018 05:51 AM - shyouhei (Shyouhei Urabe)

fix SEGV in rb_raw_obj_info()

This funcion can be called from inside of rb_ast_new().
Should add appropriate case branches.

(lldb) run
Process 9135 launched: './miniruby' (x86_64)
Process 9135 stopped

- thread #1: tid = 0xdf36b, 0x00000001000ca4f9 minirubyrb_raw_obj_info(buff="0x000000010205d158 [0] T_IMEMO", buff_size=256, obj=4328903000) + 2361 at gc.c:9617, queue = 'com.apple.main-thread', stop reason = EXC_BAD_INSTRUCTION (code=EXC_I386_INVOP, subcode=0x0) frame #0: 0x00000001000ca4f9 minirubyrb_raw_obj_info(buff="0x000000010205d158 [0] T_IMEMO", buff_size=256,

obj=4328903000) + 2361 at gc.c:9617 9614 IMEMO_NAME(iseq); 9615 IMEMO_NAME(tmpbuf); 9616 #undef IMEMO_NAME -> 9617 default: UNREACHABLE; 9618 } 9619 snprintf(buff, buff_size, "%s %s", buff, imemo_name); 9620 (lldb) bt

- thread #1: tid = 0xdf36b, 0x00000001000ca4f9 miniruby`rb_raw_obj_info(buff="0x000000010205d158 [0] T_IMEMO", buff_size=256, obj=4328903000) + 2361 at gc.c:9617, queue = 'com.apple.main-thread', stop reason = EXC_BAD_INSTRUCTION (code=EXC_I386_INVOP, subcode=0x0)
  - frame #0: 0x00000001000ca4f9 minirubyrb_raw_obj_info(buff="0x000000010205d158 [0] T_IMEMO", buff_size=256, obj=4328903000) + 2361 at gc.c:9617 frame #1: 0x00000001000c433f minirubyobj_info(obj=4328903000) + 95 at gc.c:9671 frame #2: 0x00000001000ce2ac minirubynewobj_init(klass=4302478608, flags=36890, v1=0, v2=0, v3=0, wb_protected=1, objspace=0x0000000101800410, obj=4328903000) + 444 at gc.c:1882 frame #3: 0x00000001000c0a49 minirubynewobj_of(klass=4302478608, flags=36890, v1=0, v2=0, v3=0, wb_protected=1) + 217 at gc.c:1968 frame #4: 0x00000001000c0bcb minirubyrb_imemo_new(type=imemo_ast, v1=0, v2=0, v3=0, v0=4302478608) + 75 at gc.c:2017 frame #5: 0x0000000100148f2a minirubyrb_ast_new + 58 at node.c:1118 frame #6: 0x000000010018d9e2 minirubyyycompile(vparser=4328903720, p=0x0000000100729670, fname=4328903160, line=1) + 98 at parse.y:4925 frame #7: 0x000000010018d66f minirubyparser_compile_string(vparser=4328903720, fname=4328903160, s=4328904440, line=1) + 143 at parse.y:4995 frame #8: 0x000000010018d768 minirubyrb_parser_compile_string_path(vparser=4328903720, f=4328903160, s=4328904440, line=1) + 56 at parse.y:5015 frame #9: 0x000000010018d71e minirubyrb_parser_compile_string(vparser=4328903720, f="-e", s=4328904440, line=1) + 62 at parse.y:5008 frame #10: 0x0000001002130d5 minirubyprocess_options(argc=0, argv=0x00007fff5fbfdc08, opt=0x00007fff5fbfd9e8) + 3477 at ruby.c:1754 frame #11: 0x00000001002122dd minirubyruby_process_options(argc=2, argv=0x00007fff5fbfdbf8) + 285 at ruby.c:2332 frame #12: 0x00000001000aa966 minirubyruby_options(argc=2, argv=0x00007fff5fbfdbf8) + 262 at eval.c:118 frame #13: 0x0000001000000ed4 minirubymain(argc=2, argv=0x00007fff5fbfdbf8) + 116 at main.c:42 frame #14: 0x00007fff933845ad libdyld.dylibstart + 1 frame #15: 0x00007fff933845ad libdyld.dylibstart + 1 (lldb)

Revision 65633 - 11/08/2018 09:46 AM - shyouhei (Shyouhei Urabe)

gc.c: avoid integer overflow at process exit

This is rather nitpicking but at the moment the process terminates,
heap_pages_final_slots overflows.

(lldb) bt

- thread #1: tid = 0xc0903, 0x00000001002b3bf7 miniruby`finalize_list(objspace=0x0000000101c09240, zombie=4329149840) + 999 at gc.c:2946, queue = 'com.apple.main-thread', stop reason = EXC_BAD_INSTRUCTION (code=EXC_I386_INVOP, subcode=0x0)
  - frame #0: 0x00000001002b3bf7 minirubyfinalize_list(objspace=0x0000000101c09240, zombie=4329149840) + 999 at gc.c:2946 frame #1: 0x000000010026a69e minirubyrb_objspace_call_finalizer(objspace=0x0000000101c09240) + 7118 at gc.c:3092 frame #2: 0x0000000100268ac5 minirubyrb_gc_call_finalizer_at_exit + 149 at gc.c:3015 frame #3: 0x00000001002272bc minirubyruby_finalize_1 + 156 at eval.c:146 frame #4: 0x00000001002282b6 minirubyruby_cleanup(ex=0) + 4070 at eval.c:238 frame #5: 0x0000000100228785 minirubyruby_run_node(n=0x0000000102060ad8) + 85 at eval.c:317 frame #6: 0x0000000100000b9c minirubymain(argc=2, argv=0x00007fff5fbfdc38) + 124 at main.c:42 frame #7: 0x00007fff9966a5ad libdyld.dylibstart + 1 frame #8: 0x00007fff9966a5ad libdyld.dylib`start + 1 (lldb) p objspace->heap_pages (rb_objspace::(anonymous struct)) $0 = { sorted = 0x0000000101c070b0 allocated_pages = 24 allocatable_pages = 0 sorted_length = 24 range = { [0] = 0x0000000102020028 [1] = 0x00000001020dbfd0 } freeable_pages = 0 final_slots = 0 deferred_final = 4329149840 } (lldb)

Revision 65633 - 11/08/2018 09:46 AM - shyouhei (Shyouhei Urabe)

gc.c: avoid integer overflow at process exit

This is rather nitpicking but at the moment the process terminates,
heap_pages_final_slots overflows.

(lldb) bt

- thread #1: tid = 0xc0903, 0x00000001002b3bf7 miniruby`finalize_list(objspace=0x0000000101c09240, zombie=4329149840) + 999 at gc.c:2946, queue = 'com.apple.main-thread', stop reason = EXC_BAD_INSTRUCTION (code=EXC_I386_INVOP, subcode=0x0)
  - frame #0: 0x00000001002b3bf7 minirubyfinalize_list(objspace=0x0000000101c09240, zombie=4329149840) + 999 at gc.c:2946 frame #1: 0x000000010026a69e minirubyrb_objspace_call_finalizer(objspace=0x0000000101c09240) + 7118 at gc.c:3092 frame #2: 0x0000000100268ac5 minirubyrb_gc_call_finalizer_at_exit + 149 at gc.c:3015 frame #3: 0x00000001002272bc minirubyruby_finalize_1 + 156 at eval.c:146 frame #4: 0x00000001002282b6 minirubyruby_cleanup(ex=0) + 4070 at eval.c:238 frame #5: 0x0000000100228785 minirubyruby_run_node(n=0x0000000102060ad8) + 85 at eval.c:317 frame #6: 0x0000000100000b9c minirubymain(argc=2, argv=0x00007fff5fbfdc38) + 124 at main.c:42 frame #7: 0x00007fff9966a5ad libdyld.dylibstart + 1 frame #8: 0x00007fff9966a5ad libdyld.dylib`start + 1 (lldb) p objspace->heap_pages (rb_objspace::(anonymous struct)) $0 = { sorted = 0x0000000101c070b0 allocated_pages = 24 allocatable_pages = 0 sorted_length = 24 range = { [0] = 0x0000000102020028 [1] = 0x00000001020dbfd0 } freeable_pages = 0 final_slots = 0 deferred_final = 4329149840 } (lldb)

Revision 67710 - 06/13/2019 12:23 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 7b7043e5da8589e01b94575d4ed647e909e5c875: [Backport #15793]

```
eliminate use of freed memory

rb_io_fptr_finalize_internal frees the memory region.

=================================================================
==85264==ERROR: AddressSanitizer: heap-use-after-free on address 0x610000000d8c at pc 0x5608e38077f7 bp 0x7ffe
```

```
e12d5440 sp 0x7ffee12d5438
READ of size 4 at 0x610000000d8c thread T0
    #0 0x5608e38077f6 in rb_io_memsize io.c:4749:24
    #1 0x5608e37a0481 in obj_memsize_of gc.c:3547:14
    #2 0x5608e37a4f30 in check_rvalue_consistency gc.c:1107:2
    #3 0x5608e37a2624 in RVALUE_OLD_P gc.c:1218:5
    #4 0x5608e37a5bae in rb_gc_force_recycle gc.c:6652:18
    #5 0x5608e38191f9 in rb_f_backquote io.c:9021:5
    #6 0x5608e3d8aa14 in call_cfunc_1 vm_insnhelper.c:2058:12
    #7 0x5608e3d6e23d in vm_call_cfunc_with_frame vm_insnhelper.c:2211:11
    #8 0x5608e3d54a35 in vm_call_cfunc vm_insnhelper.c:2229:12
    #9 0x5608e3d5253b in vm_call_method_each_type vm_insnhelper.c:2564:9
    #10 0x5608e3d51f50 in vm_call_method vm_insnhelper.c:2701:13
    #11 0x5608e3cf2de4 in vm_call_general vm_insnhelper.c:2734:12
    #12 0x5608e3d79918 in vm_sendish vm_insnhelper.c:3627:11
    #13 0x5608e3d06cf5 in vm_exec_core insns.def:789:11
    #14 0x5608e3d43700 in rb_vm_exec vm.c:1892:22
    #15 0x5608e3d47cbf in rb_iseq_eval_main vm.c:2151:11
    #16 0x5608e37620ca in ruby_exec_internal eval.c:262:2
    #17 0x5608e376198b in ruby_exec_node eval.c:326:12
    #18 0x5608e37617d0 in ruby_run_node eval.c:318:25
    #19 0x5608e35c9486 in main main.c:42:9
    #20 0x7f62e9421b96 in __libc_start_main /build/glibc-OTsEL5/glibc-2.27/csu/../csu/libc-start.c:310
    #21 0x5608e3522289 in _start (miniruby+0x15f289)

0x610000000d8c is located 76 bytes inside of 192-byte region [0x610000000d40,0x610000000e00)
freed by thread T0 here:
    #0 0x5608e359a2ed in free (miniruby+0x1d72ed)
    #1 0x5608e37af421 in objspace_xfree gc.c:9591:5
    #2 0x5608e37af3da in ruby_sized_xfree gc.c:9687:2
    #3 0x5608e3799ac8 in ruby_xfree gc.c:9694:5
    #4 0x5608e380746d in rb_io_fptr_finalize_internal io.c:4728:5
    #5 0x5608e38191ed in rb_f_backquote io.c:9020:5
    #6 0x5608e3d8aa14 in call_cfunc_1 vm_insnhelper.c:2058:12
    #7 0x5608e3d6e23d in vm_call_cfunc_with_frame vm_insnhelper.c:2211:11
    #8 0x5608e3d54a35 in vm_call_cfunc vm_insnhelper.c:2229:12
    #9 0x5608e3d5253b in vm_call_method_each_type vm_insnhelper.c:2564:9
    #10 0x5608e3d51f50 in vm_call_method vm_insnhelper.c:2701:13
    #11 0x5608e3cf2de4 in vm_call_general vm_insnhelper.c:2734:12
    #12 0x5608e3d79918 in vm_sendish vm_insnhelper.c:3627:11
    #13 0x5608e3d06cf5 in vm_exec_core insns.def:789:11
    #14 0x5608e3d43700 in rb_vm_exec vm.c:1892:22
    #15 0x5608e3d47cbf in rb_iseq_eval_main vm.c:2151:11
    #16 0x5608e37620ca in ruby_exec_internal eval.c:262:2
    #17 0x5608e376198b in ruby_exec_node eval.c:326:12
    #18 0x5608e37617d0 in ruby_run_node eval.c:318:25
    #19 0x5608e35c9486 in main main.c:42:9
    #20 0x7f62e9421b96 in __libc_start_main
/build/glibc-OTsEL5/glibc-2.27/csu/../csu/libc-start.c:310

previously allocated by thread T0 here:
    #0 0x5608e359a56d in malloc (miniruby+0x1d756d)
    #1 0x5608e37aed12 in objspace_xmalloc0 gc.c:9416:5
    #2 0x5608e37aebe7 in ruby_xmalloc0 gc.c:9600:12
    #3 0x5608e37aea8b in ruby_xmalloc_body gc.c:9609:12
    #4 0x5608e37a6d64 in ruby_xmalloc gc.c:11469:12
    #5 0x5608e380e4b4 in rb_io_fptr_new io.c:8040:19
    #6 0x5608e380e446 in rb_io_make_open_file io.c:8077:10
    #7 0x5608e3850ea0 in pipe_open io.c:6707:5
    #8 0x5608e384edb4 in pipe_open_s io.c:6772:12
    #9 0x5608e381910b in rb_f_backquote io.c:9014:12
    #10 0x5608e3d8aa14 in call_cfunc_1 vm_insnhelper.c:2058:12
    #11 0x5608e3d6e23d in vm_call_cfunc_with_frame vm_insnhelper.c:2211:11
    #12 0x5608e3d54a35 in vm_call_cfunc vm_insnhelper.c:2229:12
    #13 0x5608e3d5253b in vm_call_method_each_type vm_insnhelper.c:2564:9
    #14 0x5608e3d51f50 in vm_call_method vm_insnhelper.c:2701:13
    #15 0x5608e3cf2de4 in vm_call_general vm_insnhelper.c:2734:12
    #16 0x5608e3d79918 in vm_sendish vm_insnhelper.c:3627:11
    #17 0x5608e3d06cf5 in vm_exec_core insns.def:789:11
    #18 0x5608e3d43700 in rb_vm_exec vm.c:1892:22
    #19 0x5608e3d47cbf in rb_iseq_eval_main vm.c:2151:11
    #20 0x5608e37620ca in ruby_exec_internal eval.c:262:2
    #21 0x5608e376198b in ruby_exec_node eval.c:326:12
    #22 0x5608e37617d0 in ruby_run_node eval.c:318:25
    #23 0x5608e35c9486 in main main.c:42:9
```

```
    #24 0x7f62e9421b96 in __libc_start_main /build/glibc-OTsEL5/glibc-2.27/csu/../csu/libc-start.c:310

SUMMARY: AddressSanitizer: heap-use-after-free io.c:4749:24 in
rb_io_memsize
Shadow bytes around the buggy address:
  0x0c207fff8160: fa fa fa fa fa fa fa fa 00 00 00 00 00 00 00 00
  0x0c207fff8170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c207fff8180: fa fa fa fa fa fa fa fa 00 00 00 00 00 00 00 00
  0x0c207fff8190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c207fff81a0: fa fa fa fa fa fa fa fa fd fd fd fd fd fd fd fd
=>0x0c207fff81b0: fd[fd]fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c207fff81c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c207fff81d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c207fff81e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c207fff81f0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c207fff8200: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
  Shadow gap:              cc
==85264==ABORTING
```

History

#1 - 07/17/2008 06:58 AM - usa (Usaku NAKAMURA)

=begin
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

test_attr(Complex_Test) [(snip)/test/ruby/test_complex.rb:186]:
<"-0.0"> expected but was
<"0.0">.

test_fixed_bug(Complex_Test) [(snip)/test/ruby/test_complex.rb:1114]:
<"-1.0-0.0i"> expected but was
<"-1.0+0.0i">.

test_to_s(TestFloat) [(snip)/test/ruby/test_float.rb:126]:
<"1.0e+14"> expected but was
<"1.0e+014">.

test_format_float(TestSprintfComb)
[(snip)/test/ruby/test_sprintf_comb.rb:539:in block (2levels) in test_format_float'
(snip...)
(snip)/test/ruby/test_sprintf_comb.rb:526:intest_format_float']:
sprintf("%e", -123456789).
<"-1.234568e+08"> expected but was
<"-1.234568e+008">.

=end

#2 - 09/28/2008 04:40 PM - yugui (Yuki Sonoda)

- Target version set to 1.9.1 Release Candidate

=begin

=end

#3 - 10/26/2008 10:35 AM - wanabe (_ wanabe)

- File sprintf.patch added

=begin
missing/vsnprintf.c を利用するようなパッチを書きました。
vsnprintf.c 中の vsnprintf を外に出すやり方があまりうまくないように思うので
何かうまい方法をご存知の方は教えていただけるとありがたいです。
=end

#4 - 10/26/2008 03:45 PM - nobu (Nobuyoshi Nakada)

=begin
なかだです。

At Sun, 26 Oct 2008 10:35:17 +0900,
_ wanabe wrote in [ruby-dev:36932]:

> vsnprintf.c 中の vsnprintf を外に出すやり方があまりうまくないように思うので
> 何かうまい方法をご存知の方は教えていただけるとありがたいです。

win32/Makefile.subのHAVE_VSNPRINTFを消せばmissingのほうを使うよ
うになるはずです。

ちなみに、missing/vsnprintf.cへの修正はどういうものでしょうか。

--
--- 僕の前にBugはない。
--- 僕の後ろにBugはできる。
中田 伸悦

=end

#5 - 10/26/2008 08:43 PM - wanabe (_ wanabe)

=begin
ワナベです。

2008/10/26 15:45 Nobuyoshi Nakada nobu@ruby-lang.org:

> At Sun, 26 Oct 2008 10:35:17 +0900,
> _ wanabe wrote in [ruby-dev:36932]:
>
> > vsnprintf.c 中の vsnprintf を外に出すやり方があまりうまくないように思うので
> > 何かうまい方法をご存知の方は教えていただけるとありがたいです。
>
> win32/Makefile.subのHAVE_VSNPRINTFを消せばmissingのほうを使うよ
> うになるはずです。

mingw では configure で生成されたconfig.h を使うようですし
bcc では msvcrt を使わないようなので、それは最後の手段にしたいと思います。
ですがアドバイスのおかげでうまくいかない原因がわかりました。ありがとうございます。
vsnprintf を rb_win32_vsnprintf で上書きしなければいいだけだったようです。

> ちなみに、missing/vsnprintf.cへの修正はどういうものでしょうか。

説明不足でした。すみません。
test_sprintf_comb.rb が通るように修正しました。具体的には以下の通りです。

(1) sprintf("%e", -1.00000000000000000159e+100)
->修正前: "-1e+100"修正後: "-1.000000e+100"
(2) sprintf("%.0f", 0.010000000000000000208)
->修正前: "0."修正後: "0"
(3) sprintf("% #+-0.f", -0)
->修正前: "-0"修正後: "-0."
(4) sprintf("%.0G", 1)
->修正前: "1E+00"修正後: "1"

ですが今見直してみると、(1)の修正法が冗長かつ意味不明でしたので
先の rb_win32_vsnprintf の件と併せてパッチを書き直しました。

Index: include/ruby/win32.h
===
--- include/ruby/win32.h (revision 19941)
+++ include/ruby/win32.h (working copy)
@@ -243,7 +243,11 @@
extern void rb_w32_free_environ(char **);
extern int rb_w32_map_errno(DWORD);

+#if (defined(MSC_VER) && defined(_DLL)) || defined(MSVCRT_)
+#undef HAVE_VSNPRINTF
+#else
#define vsnprintf(s,n,f,l) rb_w32_vsnprintf(s,n,f,l)
+#endif
#define snprintf rb_w32_snprintf
extern int rb_w32_vsnprintf(char *, size_t, const char *, va_list);
extern int rb_w32_snprintf(char *, size_t, const char *, ...);
Index: sprintf.c
===
--- sprintf.c (revision 19941)
+++ sprintf.c (working copy)
@@ -1018,7 +1018,7 @@
need += 20;

```
    CHECK(need);
```

- sprintf(&buf[blen], fbuf, fval);

- [redacted]

```
blen += strlen(&buf[blen]);
}
break;
```

Index: numeric.c

--- numeric.c (revision 19941)
+++ numeric.c (working copy)
@@ -530,12 +530,12 @@
else if(isnan(value))
return rb_usascii_str_new2("NaN");

- sprintf(buf, "%#.15g", value); /* ensure to print decimal point */

- snprintf(buf, 32, "%#.15g", value); /* ensure to print decimal point */
if (!(e = strchr(buf, 'e'))) {
e = buf + strlen(buf);
}
if (!ISDIGIT(e[-1])) { /* reformat if ended with decimal point
(ex 111111111111111.) */

- sprintf(buf, "%#.14e", value);

- snprintf(buf, 32, "%#.14e", value);
if (!(e = strchr(buf, 'e'))) {
e = buf + strlen(buf);
}
@@ -1548,7 +1548,7 @@
char buf[24];
char *s;

- [redacted]

- [redacted]

```
if ((s = strchr(buf, ' ')) != 0) *s = '\0';
rb_raise(rb_eRangeError, "float %s out of range of integer", buf);

}
```

@@ -1694,7 +1694,7 @@
char buf[24];
char *s;

- [redacted]

- [redacted]

```
if ((s = strchr(buf, ' ')) != 0) *s = '\0';
rb_raise(rb_eRangeError, "float %s out of range of long long", buf);

}
```

Index: missing/vsnprintf.c

--- missing/vsnprintf.c (revision 19941)
+++ missing/vsnprintf.c (working copy)
@@ -753,6 +753,8 @@
#ifdef FLOATING_POINT
case 'e': /* anomalous precision */
case 'E':

- [redacted]
- [redacted]

```
    prec = (prec == -1) ?
        DEFPREC + 1 : prec + 1;
    /* FALLTHROUGH */
```

@@ -782,7 +784,7 @@
cp = cvt(_double, prec, flags, &softsign,
&expt, ch, &ndig);
if (ch == 'g' || ch == 'G') {

- [redacted]
- [redacted]

```
            ch = (ch == 'g') ? 'e' : 'E';
        else
            ch = 'g';
```

@@ -798,6 +800,8 @@
size = expt;
if (prec || flags & ALT)
size += prec + 1;

- [redacted]
- [redacted]

```
        } else  /* "0.X" */
            size = prec + 2;
    } else if (expt >= ndig) {  /* fixed g fmt */
```

@@ -1008,13 +1012,15 @@
if (ch >= 'f') { /* 'f' or 'g' /
if (_double == 0) {
/ kludge for __dtoa irregularity */

- [redacted]
- [redacted]

```
                (flags & ALT) == 0) {
                PRINT("0", 1);
            } else {
                PRINT("0.", 2);
                PAD(ndig - 1, zeroes);
            }
```

- [redacted]
- [redacted]

```
        } else if (expt <= 0) {
            PRINT("0.", 2);
            PAD(-expt, zeroes);
```

--
[illegible]

=end

#6 - 10/26/2008 10:38 PM - nobu (Nobuyoshi Nakada)

=begin
なかだです。

At Sun, 26 Oct 2008 20:42:54 +0900,
wanabe wrote in [ruby-dev:36935]:

> win32/Makefile.subのHAVE_VSNPRINTFを消せばmissingのほうを使うようになるはずです。

> mingw では configure で生成されたconfig.h を使うようですし
> bcc では msvcrt を使わないようなので、それは最後の手段にしたいと思います。

mingwではconfiugre.inの ac_cv_func_vsnprintf=yes という行を消せばできるはずです。bccはbcc32/Makefile.subにある定義を使うので、win32/Makefile.subを変更しても影響は受けません。

> ですがアドバイスのおかげでうまくいかない原因がわかりました。ありがとうございます。
> vsnprintf を rb_win32_vsnprintf で上書きしなければいいだけだったようです。

rb_w32_vsnprintf()はmsvcrtのvsnprintf()のバグを回避するためのものなので、missingのものを使うのであればsnprintf()も不要になります。バイナリ互換についてはwin32/mkexports.rbでaliasしておけば充分でしょう。

> ちなみに、missing/vsnprintf.cへの修正はどういうものでしょうか。

> 説明不足でした。すみません。
> test_sprintf_comb.rb が通るように修正しました。具体的には以下の通りです。

なるほど。

- sprintf(buf, "%#.15g", value); /* ensure to print decimal point */
- snprintf(buf, 32, "%#.15g", value); /* ensure to print decimal point */

sizeof(buf)のほうがいいと思います。

--
--- 僕の前にBugはない。
--- 僕の後ろにBugはできる。
中田 伸悦

=end

#7 - 12/21/2008 12:12 AM - yugui (Yuki Sonoda)

- Due date set to 12/24/2008

=begin

=end

#8 - 12/22/2008 10:29 AM - yugui (Yuki Sonoda)

- Assignee changed from usa (Usaku NAKAMURA) to nobu (Nobuyoshi Nakada)

- Priority changed from 3 to 5

=begin
「分かってるならコミットして」(うささん)とのことなのでなかださんに任せます。
=end

#9 - 12/22/2008 12:17 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

=begin

Applied in changeset r20910.
=end

#10 - 02/26/2010 10:32 PM - mame (Yusuke Endoh)

- Target version changed from 1.9.1 Release Candidate to 1.9.2

- ruby -v set to ruby 1.9.2dev (2010-02-26) [i386-mingw32]

=begin
遠藤です。

古いチケットですが reopen します。

> test_to_s(TestFloat) [(snip)/test/ruby/test_float.rb:126]:
> <"1.0e+14"> expected but was
> <"1.0e+014">.

mingw32 で今でも再現するようです。
mingw では直ってなかったのか regression なのかは知りません。

$ ./ruby test/ruby/test_float.rb
Loaded suite test/ruby/test_float
Started
.....F..
Finished in 1.019000 seconds.

1) Failure:
test_to_s(TestFloat) [test/ruby/test_float.rb:126]:
<"1.0e+18"> expected but was
<"1.0e+018">.

31 tests, 1120 assertions, 1 failures, 0 errors, 0 skips

--
Yusuke Endoh mame@tsg.ne.jp
=end

#11 - 02/27/2010 10:40 PM - wanabe (_ wanabe)

- File force_use_missing.patch added

=begin
こんな感じでどうでしょうか。
configure.in は触ったことがないのでこれで良いのかよくわからないのですが
問題なさそうでしたらこれでコミットしたいと思います。
=end

#12 - 02/28/2010 10:50 AM - mame (Yusuke Endoh)

=begin
遠藤です。

2010年2月27日22:40 _ wanabe redmine@ruby-lang.org:

> こんな感じでどうでしょうか。
> configure.in は触ったことがないのでこれで良いのかよくわからないのですが
> 問題なさそうでしたらこれでコミットしたいと思います。

mingw で直ることを確認しました。

$ ./ruby test/ruby/test_float.rb
Loaded suite test/ruby/test_float
Started
.....
Finished in 1.019000 seconds.

31 tests, 1120 assertions, 0 failures, 0 errors, 0 skips

ありがとうございます。

--
Yusuke ENDOH mame@tsg.ne.jp

=end

Files

sprintf.patch	4.24 KB	10/26/2008	wanabe (_ wanabe)
force_use_missing.patch	882 Bytes	02/27/2010	wanabe (_ wanabe)