Ruby master - Bug #15358
Segfault in rb_transient_heap_mark when running Sequel's specs
11/28/2018 08:54 PM - jeremyevans0 (Jeremy Evans)

Status: Closed
Priority: Normal
Assignee: ko1 (Koichi Sasada)
Target version: 2.6
ruby -v:
ruby 2.6.0preview3 (2018-11-06 trunk 65578) [x86_64-openbsd]
Backport: 2.4: UNKNOWN, 2.5: UNKNOWN

Description
When running Sequel's plugin specs, there is a segfault that is dependent on execution order. I have been seeing it on OpenBSD after upgrading from ruby-2.6.0-preview2 to ruby-2.6.0-preview3, and I was not sure if it was OpenBSD specific. It has now occurred on Travis running Linux (https://travis-ci.org/jeremyevans/sequel/jobs/460952288), so this appears to be a general issue and not specific to OpenBSD.

This error is not caused by a C extensions. The only C extension loaded in Sequel's plugin specs is nokogiri, and I have verified that this error occurs even if nokogiri is not loaded. Sequel's plugin specs pass on ruby 1.9.3 to 2.6.0-preview2, and JRuby 1.7 to 9.2, so this is unlikely to be a problem inside Sequel.

Here's the backtrace when running the specs inside gdb:

(gdb) bt
#0 0x0000088014bd213 in rb_transient_heap_mark (obj=9347566859680, ptr=0x0) at transient_heap.c:525
#1 0x0000088014bdefec in mark_hash (objspace=0x8803bc38800, hash=9347566859680) at gc.c:4267
#2 0x0000088014bce131 in gc_mark_children (objspace=0x8803bc38800, obj=9347566859680) at gc.c:478
#3 0x0000088014bcc753 in gc_mark_stacked_objects (objspace=0x8803bc38800, incremental=0, count=0) at gc.c:4883
#4 0x0000088014bca165 in gc_mark_stacked_objects_all (objspace=0x8803bc38800) at gc.c:4923
#5 0x0000088014bcb8e4 in gc_marks (objspace=0x8803bc38800, full_mark=0) at gc.c:5863
#6 0x0000088014bca165 in gc_marks_rest (objspace=0x8803bc38800) at gc.c:5803
#7 0x0000088014bca165 in gc_start (objspace=0x8803bc38800, reason=256) at gc.c:6665
#8 0x0000088014bca165 in heap_prepare (objspace=0x8803bc38800, heap=0x8803bc38820) at gc.c:1740
#9 0x0000088014bca165 in heap_get_freeobj_from_next_freepage (objspace=0x8803bc38800, heap=0x8803bc38820) at gc.c:1752
#10 0x0000088014bca165 in heap_get_freeobj (objspace=0x8803bc38800, heap=0x8803bc38820) at gc.c:1789
#11 0x0000088014bca165 in newobj_slowpath (klass=0, flags=24602, v1=0, v2=258601, v3=9346992938920, objspace=0x8803bc38800, wb_protected=1) at gc.c:1919
#12 0x0000088014bca165 in newobj_slowpath_wb_protected (klass=0, flags=24602, v1=0, v2=258601, v3=9346992938920, objspace=0x8803bc38800, wb_protected=1) at gc.c:1931
#13 0x0000088014bca165 in newobj_of (klass=0, flags=24602, v1=0, v2=258601, v3=9346992938920, wb_protected=1) at gc.c:1967
#14 0x0000088014bca165 in rb_imemo_new (type=imemo_ment, v1=0, v2=258601, v3=9346992938920) at gc.c:2010
#15 0x0000088014bca165 in rb_method_entry_alloc (called_id=258601, owner=9346992938920, defined_class=0, def=0x0) at vm_method.c:369
#16 0x0000088014bca165 in rb_method_entry_create (called_id=258601, klass=9346992938920, visi=METH_VISI_PUBLIC, def=0x0) at vm_method.c:390
#17 0x0000088014bca165 in rb_method_entry_make (klass=9346992938920, mid=258601, defined_class=9346992938920, visi=METH_VISI_PUBLIC, type=VM_METHOD_TYPE_ISEQ, def=0x0, original_id=258601, opts=0x7f7fff87d0) at vm_method.c:597
#18 0x0000088014bca165 in rb_defmeth (klass=9346992938920, mid=258601, type=VM_METHOD_TYPE_ISEQ, owner=9346992938920, defined_class=0, def=0x0) at vm_method.c:653
#19 0x0000088014bca165 in rb_defmeth (klass=9346992938920, mid=258601, iseq=0x880443190a8, cref=0x88044319350, visi=METH_VISI_PUBLIC) at vm_method.c:672
#20 0x0000088014bca165 in rb_define_method (obj=8, id=258601, iseqval=9346992935080, is_singleton=0) at vm.c:2509
#21 0x0000088014bca165 in m_core_defmethod (self=9347027151000, sym=66201868, iseqval=9346992935080) at vm.c:2629

03/16/2022 1/5
Based on the backtrace, the issue appears to be in the new transient heap code added in preview3, specifically that `rb_transient_heap_mark` is called with a NULL pointer as the second argument. If any additional information is needed from gdb, please let me know and I can update with more information.

Currently, to reproduce:

```
gem install --development sequel # install dependencies
git clone git@github.com:jeremyevans/sequel.git
cd sequel
git checkout a0203701957de236d4da06a13ced6a89c21e0f5d
ruby spec/plugin_spec.rb --seed 59498
```

I will try to work on a smaller reproducer if that would be helpful.

**Associated revisions**

Revision a763e346 - 11/29/2018 08:03 AM - ko1 (Koichi Sasada)

clear dst Hash on Hash#replace. [Bug #15358]

- hash.c (linear_copy): solve two issues on Hash#replace. (1) fix memory leak (1-1) don't allocate memory if destination already has a memory area. (1-2) free destination memory if src is NULL. (2) clear transient heap flag if src is NULL. [Bug #15358]

```
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@66091 b2dd03c8-39d4-4d8f-98ff-823fe69b080e
Revision 66091 - 11/29/2018 08:03 AM - ko1 (Koichi Sasada)

clear dst Hash on Hash#replace. [Bug #15358]

- hash.c (linear_copy): solve two issues on Hash#replace. (1) fix memory leak (1-1) don't allocate memory if destination already has a memory area. (1-2) free destination memory if src is NULL. (2) clear transient heap flag if src is NULL. [Bug #15358]

Revision 66091 - 11/29/2018 08:03 AM - ko1 (Koichi Sasada)

clear dst Hash on Hash#replace. [Bug #15358]

- hash.c (linear_copy): solve two issues on Hash#replace. (1) fix memory leak (1-1) don't allocate memory if destination already has a memory area. (1-2) free destination memory if src is NULL. (2) clear transient heap flag if src is NULL. [Bug #15358]

**History**

#1 - 11/29/2018 07:43 AM - ko1 (Koichi Sasada)

Thank you for your report. I found a bug and will commit soon.
Thanks,
Koichi

#2 - 11/29/2018 08:04 AM - ko1 (Koichi Sasada)
- Status changed from Open to Closed

Applied in changeset trunk|r66091.

clear dst Hash on Hash#replace. [Bug #15358]

- hash.c (linear_copy): solve two issues on Hash#replace. (1) fix memory leak   (1-1) don't allocate memory if destination already   has a   memory area. (1-2) free destination memory if src is NULL. (2) clear transient heap flag if src is NULL. [Bug #15358]