Ruby master - Bug #18516

Memory leak on aliasing method to itself

01/26/2022 01:22 PM - ibylich (Ilya Bylich)

Status: Closed
Priority: Normal
Assignee:
Target version:
ruby -v: ruby 3.2.0dev
Backport: 2.6: WONTFIX, 2.7: REQUIRED, 3.0: REQUIRED, 3.1: DONE

Description

The following code produces a memory leak:

```ruby
class A
  1.upto(Float::INFINITY) do |i|
    define_method("foo_#{i}") {}

    alias :"foo_#{i}" :"foo_#{i}"

    remove_method :"foo_#{i}" end
end
```

It is very artificial, but it's required for LSAN/Valgrind integration.

The reason why it leaks is the following:

- alias foo foo increments alias_count even if no alias is created
- later during GC rb_free_method_entry is called that in turn calls rb_method_definition_release that calls free only if alias_count + complemented_count == 0.
- In this particular case alias_count is 1, and so no cleanup happens.

I've been asked to create tickets for memory leaks that I find during LSAN integration if the leak affects previous versions of Ruby. From what I see it does.

Possible fix: https://github.com/ruby/ruby/pull/5492. But like I mentioned in PR I'm not sure if it's correct, nobu (Nobuyoshi Nakada) mentioned in my main PR aliasing method to itself is used to swallow some warning on method redefinition, but I don't what's the case there.

Associated revisions

Revision e89d8070 - 01/27/2022 06:46 AM - nobu (Nobuyoshi Nakada)

Fix memory leak at the same named alias [Bug #18516]

When aliasing a method to the same name method, set a separate bit flag on that method definition, instead of the reference count increment. Although this kind of alias has no actual effect at runtime, is used as the hack to suppress the method re-definition warning.

Revision 42c9ef76 - 02/02/2022 11:04 PM - naruse (Yui NARUSE)

merge revision(s) 7ff1bf317887c0d7b21e91ad548d07b9059c540c,e89d80702bd98a8276243a7fcaa2a158b3bb659: [Backport #18516]

An alias can suppress method redefinition warning

```diff
---
test/ruby/test_alias.rb | 11 ++++++++++++
1 file changed, 11 insertions(+)
```

Fix memory leak at the same named alias [Bug #18516]

When aliasing a method to the same name method, set a separate bit flag on that method definition, instead of the reference count

03/12/2022
increment. Although this kind of alias has no actual effect at runtime, is used as the hack to suppress the method re-definition warning.

---
method.h | 1 +
test/ruby/test_alias.rb | 18 ++++++++++++++++++
vm_method.c | 9 ++++++++-
3 files changed, 27 insertions(+), 1 deletion(-)

History

#1 - 01/26/2022 02:30 PM - ibylich (Ilya Bylich)
As nobu (Nobuyoshi Nakada) mentioned the following code gives no warnings (in verbose mode):

class A
def foo; end
alias foo foo;
def foo; end
end

but if we stop increment alias_count on alias foo foo the warning appears (because alias foo foo becomes a real noop, with absolutely no side effect).

#2 - 01/26/2022 03:20 PM - ufuk (Ufuk Kayserilioglu)
ibylich (Ilya Bylich) wrote in #note-1:

but if we stop increment alias_count on alias foo foo the warning appears (because alias foo foo becomes a real noop, with absolutely no side effect).

Rails heavily depends on this behaviour and if this is changed it will end up raising a lot of warnings. I am not even sure if there is another way to replace methods without raising any warnings, if this is taken away.

#3 - 01/26/2022 03:44 PM - nobu (Nobuyoshi Nakada)
https://github.com/ruby/ruby/pull/5493

#4 - 01/27/2022 06:46 AM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed

Applied in changeset git|e89d80702bd98a8276243a7fcaa2a158b3fb659.

Fix memory leak at the same named alias [Bug #18516]

When aliasing a method to the same name method, set a separate bit flag on that method definition, instead of the reference count increment. Although this kind of alias has no actual effect at runtime, is used as the hack to suppress the method re-definition warning.

#5 - 01/27/2022 09:11 AM - byroot (Jean Boussier)
- Backport changed from 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN to 2.6: WONTFIX, 2.7: REQUIRED, 3.0: REQUIRED, 3.1: REQUIRED

I took the liberty to set the backport field.

I was able to repro on 2.6, 2.7,3.0 and 3.1. However 2.6 is security only, I don't think it qualifies.

#6 - 02/02/2022 11:05 PM - naruse (Yui NARUSE)
- Backport changed from 2.6: WONTFIX, 2.7: REQUIRED, 3.0: REQUIRED, 3.1: REQUIRED to 2.6: WONTFIX, 2.7: REQUIRED, 3.0: REQUIRED, 3.1: DONE

ruby_3.1 42c9ef769f210d88241a114395dd5fc27b2fb87 merged revision(s)
7ff1bf317887c0d7b21e91ad548d07b9f05c540c,e89d80702bd98a8276243a7fcaa2a158b3fb659.