

Ruby trunk - Bug #14475

String de-duplication is broken in files with frozen_string_literal: true

02/14/2018 07:37 AM - sam.saffron (Sam Saffron)

Status: Closed	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: 2.5.0, 2.6.0	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN
Description Create 2 files: test.rb <pre>\$LOAD_PATH.unshift(File.dirname(__FILE__)) require 'frozen' puts frozen.object_id puts frozen.object_id</pre> frozen.rb <pre># frozen_string_literal: true A = "a" def frozen -"#{A}" end</pre> Run test.rb <pre>% ruby test.rb 70189973179400 70189973179260</pre> Change to # frozen_string_literal: false And you get <pre>70189973181360 70189973181360</pre> So something is over-optimising here, fix should be backported to 2.5.0 imo.	
Related issues: Related to Ruby trunk - Feature #14478: String #uminus should de-dupe uncondi... Closed	

Associated revisions

Revision 7606806c - 02/14/2018 10:35 PM - normal

compile.c: drop freezestring insn on String#-@

Followup to r62039 and remove the redundant freezestring insn which was preventing deduplication from String#-@

- compile.c (iseq_peephole_optimize): drop freezestring insn on String#-@ [ruby-core:85542] [Bug #14475]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62407 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 62407 - 02/14/2018 10:35 PM - normalperson (Eric Wong)

compile.c: drop freezestring insn on String#-@

Followup to r62039 and remove the redundant freeze string insns which were preventing deduplication from String#-@

- compile.c (iseq_peephole_optimize): drop freeze string insns on String#-@ [ruby-core:85542] [Bug #14475]

Revision 62407 - 02/14/2018 10:35 PM - normal

compile.c: drop freeze string insns on String#-@

Followup to r62039 and remove the redundant freeze string insns which were preventing deduplication from String#-@

- compile.c (iseq_peephole_optimize): drop freeze string insns on String#-@ [ruby-core:85542] [Bug #14475]

Revision 86de3e41 - 02/16/2018 01:15 AM - nobu (Nobuyoshi Nakada)

compile.c: keep debug info

- compile.c (iseq_peephole_optimize): keep freeze string insns with debug info. [Bug #14475]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62418 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 62418 - 02/16/2018 01:15 AM - nobu (Nobuyoshi Nakada)

compile.c: keep debug info

- compile.c (iseq_peephole_optimize): keep freeze string insns with debug info. [Bug #14475]

Revision 62418 - 02/16/2018 01:15 AM - nobu (Nobuyoshi Nakada)

compile.c: keep debug info

- compile.c (iseq_peephole_optimize): keep freeze string insns with debug info. [Bug #14475]

History

#1 - 02/14/2018 08:40 PM - sam.saffron (Sam Saffron)

Simpler repro:

```
x = "#{1}".freeze
p x.object_id
x = -x
p x.object_id
```

```
x = "#{1}".freeze
p x.object_id
x = -x
p x.object_id
```

```
70210401536460
70210401536460
70210401536240
70210401536240
```

#2 - 02/14/2018 08:43 PM - sam.saffron (Sam Saffron)

Workaround ... whenever de-duping strings always use -+

```
x = "#{1}".freeze
# this is always properly de-duped
x = -+x
```

Any objections to changing:

```
static VALUE
str_uminus(VALUE str)
{
    if (OBJ_FROZEN(str)) {
        return str;
    }
    else {
        return rb_fstring(str);
    }
}
```

To

```
static VALUE
str_uminus(VALUE str)
{
  return rb_fstring(str);
}
```

#3 - 02/14/2018 10:12 PM - normalperson (Eric Wong)

sam.saffron@gmail.com wrote:

Workaround ... whenever de-duping strings always use --+

Yes, String#-@ documentation states:

"If the string is frozen, then return the string itself."

So I guess it is spec...

However, it seems compile-time can skip freezestring instructions for -@ like it does with +@ since [r62039](#)

```
diff --git a/compile.c b/compile.c
index 5bca0dca99..7d065b25d3 100644
--- a/compile.c
+++ b/compile.c
@@ -2851,11 +2851,11 @@ iseq_peephole_optimize(rb_iseq_t *iseq, LINK_ELEMENT *list, const int do_tailcal
  struct rb_call_info *ci = (struct rb_call_info *)OPERAND_AT(niobj, 0);
  /*
   * freezestring debug_info
-  * send <:+@, 0, ARG_SIMPLE>
+  * send <:+@, 0, ARG_SIMPLE> # :-@, too
   * =>
-  * send <:+@, 0, ARG_SIMPLE>
+  * send <:+@, 0, ARG_SIMPLE> # :-@, too
   */
-  if (ci->mid == idUPlus &&
+  if ((ci->mid == idUPlus || ci->mid == idUMinus) &&
      (ci->flag & VM_CALL_ARGS_SIMPLE) &&
      ci->orig_argc == 0) {
    ELEM_REMOVE(list);
diff --git a/test/ruby/test_optimization.rb b/test/ruby/test_optimization.rb
index 6e463e1863..55e7acbbf0 100644
--- a/test/ruby/test_optimization.rb
+++ b/test/ruby/test_optimization.rb
@@ -567,6 +567,21 @@ def test_peephole_string_literal_range
end
end

+ def test_peephole_dstr
+   code = "#{<<~'begin;'}\n#{<<~'end;'}"
+   begin;
+     exp = ('a').object_id
+     z = 'a'
+     exp == ("#{z}").object_id
+   end;
+   [ false, true ].each do |fsl|
+     iseq = RubyVM::InstructionSequence.compile(code,
+                                               frozen_string_literal: fsl)
+     assert_equal(true, iseq.eval,
+                  "[ruby-core:85542] [Bug #14475] fsl: #{fsl}")
+   end
+ end
+
def test_branch_condition_backquote
  bug = '[ruby-core:80740] [Bug #13444] redefined backquote should be called'
  class << self
```

#4 - 02/14/2018 10:27 PM - sam.saffron (Sam Saffron)

"If the string is frozen, then return the string itself."

Yeah I do not agree with this documentation, I think it should be changed.

Trouble is that there is no simple way to de-duplicate unconditionally without inefficiency:

Say x is an arbitrary string (either frozen or unfrozen) :

```
x = -x # may return a non fstring
x = -+x # will return fstring, but makes an unneeded copy
x = -x.dup # fstring again, unneeded copy
x = x.frozen? ? -+x : -x # too verbose, unneeded copy
```

Instead why not change it so - is deduped unconditionally?

#5 - 02/14/2018 10:35 PM - Anonymous

- Status changed from Open to Closed

Applied in changeset [trunk|r62407](https://github.com/ruby/ruby/commit/trunk|r62407).

compile.c: drop freezestring insn on String#-@

Followup to [r62039](https://bugs.ruby-lang.org/issues/14475) and remove the redundant freezestring insn which was preventing deduplication from String#-@

- compile.c (iseq_peephole_optimize): drop freezestring insn on String#-@ [ruby-core:85542] [Bug [#14475](https://bugs.ruby-lang.org/issues/14475)]

#6 - 02/14/2018 10:45 PM - sam.saffron (Sam Saffron)

Opened <https://bugs.ruby-lang.org/issues/14478> to discuss change to uminus

#7 - 02/14/2018 10:52 PM - normalperson (Eric Wong)

sam.saffron@gmail.com wrote:

"If the string is frozen, then return the string itself."

Yeah I do not agree with this documentation, I think it should be changed.

Trouble is that there is no simple way to de-duplicate unconditionally without either inefficiency or side effects:

Yeah, I originally proposed a new method back in the day but compromised with String#-@ <https://bugs.ruby-lang.org/issues/13077>

Say x is an arbitrary string (either frozen or unfrozen) :

```
x = -x # may return a non fstring
x = -+x # will return fstring, but makes an unneeded copy
x = -x.dup # fstring again, unneeded copy
```

Maybe these can be optimized in the VM (still ugly, and redefinition checks aren't free, either).

```
x = x.frozen? ? -+x : -x # too verbose
```

Instead why not change it so `` deduped unconditionally?

At this point, it may break existing code :<

#8 - 02/14/2018 11:42 PM - sam.saffron (Sam Saffron)

I created a new topic for this at: <https://bugs.ruby-lang.org/issues/14478>

At this point, it may break existing code :<

I think it is worth the risk cause adding #fstring here seem overkill and almost nobody would be using uminus now anyway cause they do not know about it.

#9 - 02/15/2018 01:39 AM - duerst (Martin Dürst)

- Related to Feature #14478: String #uminus should de-dupe unconditionally added