

Ruby trunk - Feature #13295

[PATCH] compile.c: apply opt_str_freeze to String#-@ (uminus)

03/08/2017 09:26 PM - normalperson (Eric Wong)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
Description	
The same optimization used for "literal string".freeze can easily apply to uminus without introducing any compatibility problems.	
I can't think of any problems this would introduce, but I'll wait a few days for feedback before committing.	

Associated revisions

Revision 833e42d1 - 03/26/2017 10:21 AM - normal

NEWS: feature number for String#-@ compile-time optimization

[ruby-core:79971] [Feature #13295]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@58142 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 58142 - 03/26/2017 10:21 AM - normalperson (Eric Wong)

NEWS: feature number for String#-@ compile-time optimization

[ruby-core:79971] [Feature #13295]

Revision 58142 - 03/26/2017 10:21 AM - normal

NEWS: feature number for String#-@ compile-time optimization

[ruby-core:79971] [Feature #13295]

Revision 58142 - 03/26/2017 10:21 AM - normal

NEWS: feature number for String#-@ compile-time optimization

[ruby-core:79971] [Feature #13295]

History

#1 - 03/13/2017 03:28 PM - shyouhei (Shyouhei Urabe)

We looked at this issue in today's developer meeting.

Koichi pointed out that you have to consider redefinition of String#-@. You need to take care about such situation, like we do already for #freeze.

#2 - 03/26/2017 06:21 AM - normalperson (Eric Wong)

shyouhei@ruby-lang.org wrote:

We looked at this issue in today's developer meeting.

Koichi pointed out that you have to consider redefinition of String#-@. You need to take care about such situation, like we do already for #freeze.

Oops, I already committed r57828 before this message. opt_str_freeze insn already takes care of redefinition in insns.def; so extra check for compile.c won't be affected.

#3 - 03/26/2017 10:21 AM - Anonymous

- Status changed from Open to Closed

Applied in changeset [trunk|r58142](#).

NEWS: feature number for String#-@ compile-time optimization

[ruby-core:79971] [Feature [#13295](#)]

#4 - 03/27/2017 05:41 AM - ko1 (Koichi Sasada)

On 2017/03/26 15:16, Eric Wong wrote:

Oops, I already committed r57828 before this message.
opt_str_freeze insn already takes care of redefinition in
insns.def; so extra check for compile.c won't be affected.

now it checks only 'freeze' redefinition (*1). I'll fix it before 2.5
release (maybe with new a instruction (*2), which replace existing
opt_str_freeze).

```
*1:
class String
  def -@
  p :redef
  self
  end
end

p -'foo'

*2:
opt_str_freeze "foo"
->
opt_frozen "foo", label # jump if String#freeze or String#-@
# are not redefined
putstring "foo"
send "-@"
label:
```

(or something smarter insn)

```
--
// SASADA Koichi at atdot dot net
```

#5 - 03/27/2017 06:08 AM - normalperson (Eric Wong)

SASADA Koichi ko1@atdot.net wrote:

On 2017/03/26 15:16, Eric Wong wrote:

Oops, I already committed r57828 before this message.
opt_str_freeze insn already takes care of redefinition in
insns.def; so extra check for compile.c won't be affected.

now it checks only 'freeze' redefinition (*1). I'll fix it before 2.5
release (maybe with new a instruction (*2), which replace existing
opt_str_freeze).

You're absolutely right; sorry about that. I will fix simply, for now.

But maybe you can do (*2) later.

#6 - 03/27/2017 06:41 AM - ko1 (Koichi Sasada)

On 2017/03/27 14:55, Eric Wong wrote:

You're absolutely right; sorry about that. I will fix simply, for now.

Nobody redef them so no problem :p

--
// SASADA Koichi at atdot dot net

Files

0001-compile.c-apply-opt_str_freeze-to-String-uminus.patch	1.47 KB	03/08/2017	normalperson (Eric Wong)
--	---------	------------	--------------------------