prepend した Class/Module が継承ツリー上にある Class/Module で ancestors に Class が入り込む

06/28/2012 10:08 AM - nagachika (Tomoyuki Chikanaga)

Status: Closed
Priority: Normal
Assignee: 
Target version: 2.0.0

ruby -v:
ruby 2.0.0dev (2012-06-28 trunk 36243)
x86_64-darwin11.4.0

Description
prepend した Class/Module で Module#ancestors の順番や Class が入り込む問題は r36241 で対処されましたが
まだ ancestors を呼ぶ Class/Module そのものではなくてその superclass や include している Module で
prepend を使っていると順番が include と同じになる & Class が入り込むようです。

module M; end
class C1
prepend M
end
class C2 < C1
end
C2.ancestors => [C2, C1, M, Class, Object, Kernel, BasicObject]

Associated revisions
Revision edb1fc4e - 07/02/2012 08:06 AM - nobu (Nobuyoshi Nakada)
prepend: fix mixing with include

* class.c (rb_include_module): include modules after the origin.
* class.c (include_modules_at): skip prepended modules.
* class.c (rb_prepend_module): now basic.klass in ICLASS refers the old original class/module. [ruby-dev:45868][Bug #6662]
* class.c (rb_mod_ancestors): ditto.
* vm_method.c (search_method): search method entry from the origin iclass.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36266 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 36266 - 07/02/2012 08:06 AM - nobu (Nobuyoshi Nakada)
prepend: fix mixing with include

* class.c (rb_include_module): include modules after the origin.
* class.c (include_modules_at): skip prepended modules.
* class.c (rb_prepend_module): now basic.klass in ICLASS refers the old original class/module. [ruby-dev:45868][Bug #6662]
* class.c (rb_mod_ancestors): ditto.
* vm_method.c (search_method): search method entry from the origin iclass.

Revision 36266 - 07/02/2012 08:06 AM - nobu (Nobuyoshi Nakada)
prepend: fix mixing with include

* class.c (rb_include_module): include modules after the origin.
* class.c (include_modules_at): skip prepended modules.
* class.c (rb_prepend_module): now basic.klass in ICLASS refers the old original class/module. [ruby-dev:45868][Bug #6662]
* class.c (rb_mod_ancestors): ditto.
* vm_method.c (search_method): search method entry from the origin iclass.

Revision 36266 - 07/02/2012 08:06 AM - nobu (Nobuyoshi Nakada)
prepend: fix mixing with include

* class.c (rb_include_module): include modules after the origin.
* class.c (include_modules_at): skip prepended modules.
* class.c (rb_prepend_module): now basic.klass in ICLASS refers the old original class/module. [ruby-dev:45868][Bug #6662]
* class.c (rb_mod_ancestors): ditto.
* vm_method.c (search_method): search method entry from the origin iclass.
prepend: fix mixing with include

- class.c (rb_include_module): include modules after the origin.
- class.c (include_modules_at): skip prepended modules.
- class.c (rb_prepend_module): now basic.klass in ICLASS refers the old original class/module. [ruby-dev:45868][Bug #6662]
- class.c (rb_mod_ancestors): ditto.
- vm_method.c (search_method): search method entry from the origin iclass.

Revision 36266 - 07/02/2012 08:06 AM - nobu (Nobuyoshi Nakada)
prepend: fix mixing with include

- class.c (rb_include_module): include modules after the origin.
- class.c (include_modules_at): skip prepended modules.
- class.c (rb_prepend_module): now basic.klass in ICLASS refers the old original class/module. [ruby-dev:45868][Bug #6662]
- class.c (rb_mod_ancestors): ditto.
- vm_method.c (search_method): search method entry from the origin iclass.

Revision 36266 - 07/02/2012 08:06 AM - nobu (Nobuyoshi Nakada)
prepend: fix mixing with include

- class.c (rb_include_module): include modules after the origin.
- class.c (include_modules_at): skip prepended modules.
- class.c (rb_prepend_module): now basic.klass in ICLASS refers the old original class/module. [ruby-dev:45868][Bug #6662]
- class.c (rb_mod_ancestors): ditto.
- vm_method.c (search_method): search method entry from the origin iclass.

History

#1 - 07/02/2012 05:06 PM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r36266.
Tomoyuki, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.