Ruby master - Bug #6048

{Unbound}Method#hash doesn't always return the right value

02/20/2012 03:19 PM - marcandre (Marc-Andre Lafortune)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>marcandre (Marc-Andre Lafortune)</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.0.0</td>
</tr>
<tr>
<td>ruby -v:</td>
<td>r34632</td>
</tr>
<tr>
<td>Backport:</td>
<td></td>
</tr>
</tbody>
</table>

Description

{Unbound}Method#hash doesn't always return the right value.

map, collect = Array.instance_method(:map), Array.instance_method(:collect)

map.eql?(collect)        # => true
map.hash == collect.hash # => false

I'm tempted to think that this is an obvious bug with an obvious solution but let me state:

As per the documentation and the design of hash tables, if two objects are eql? then they must have the same hash.

Either map should not be eql? to collect or else their hash should be the same.

As they are identical methods, it is correct that they are eql?, so hash must return the same value.

My proposed behavior passes my strict superiority test and is also "straightforward" as I could find no intent for the current behavior.

One solution is to ensure that all aliased methods are defined using rb_define_alias, which appears to yield the same hash. Another is to compute the hash correctly. Maybe there is a third approach.

I'm not super confident I took the right approach, but here is a patch for the second one. I think it is more robust and more consistent with how {Unbound}Method.eql? is implemented. It may also fix other cases of mismatch between eql? and hash, I didn't investigate further.

I'd be grateful if someone could review the patch (Koichi?) and let me know if I took the right approach and if I put things in the right place.

Thanks

Marc-André

Associated revisions

Revision f12c14d3 - 02/21/2012 12:13 AM - marcandre (Marc-Andre Lafortune)

- proc.c (method_hash, proc_hash): Fix {Unbound}Method#hash
  [Bug #6048]. Isolate hash computation for proc
- internal.h: Declaration for above
- vm_method.c (rb_method_definition_hash): Computation for hash part of a method definition
- method.h: Declaration for above
- test/ruby/test_method.rb: Test for above

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@34715 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 34715 - 02/21/2012 12:13 AM - marcandre (Marc-Andre Lafortune)
proc.c (method_hash, proc_hash): Fix [Unbound]Method#hash
[Bug #6048]. Isolate hash computation for proc

internal.h: Declaration for above

vm_method.c (rb_method_definition_hash): Computation for
hash part of a method definition

method.h: Declaration for above

test/ruby/test_method.rb: Test for above

Revision 34715 - 02/21/2012 12:13 AM - marcandre (Marc-Andre Lafortune)

proc.c (method_hash, proc_hash): Fix [Unbound]Method#hash
[Bug #6048]. Isolate hash computation for proc

internal.h: Declaration for above

vm_method.c (rb_method_definition_hash): Computation for
hash part of a method definition

method.h: Declaration for above

test/ruby/test_method.rb: Test for above

Revision 34715 - 02/21/2012 12:13 AM - marcandre (Marc-Andre Lafortune)

proc.c (method_hash, proc_hash): Fix [Unbound]Method#hash
[Bug #6048]. Isolate hash computation for proc

internal.h: Declaration for above

vm_method.c (rb_method_definition_hash): Computation for
hash part of a method definition

method.h: Declaration for above

test/ruby/test_method.rb: Test for above

Revision 34715 - 02/21/2012 12:13 AM - marcandre (Marc-Andre Lafortune)

proc.c (method_hash, proc_hash): Fix [Unbound]Method#hash
[Bug #6048]. Isolate hash computation for proc

internal.h: Declaration for above

vm_method.c (rb_method_definition_hash): Computation for
hash part of a method definition

method.h: Declaration for above

test/ruby/test_method.rb: Test for above

Revision 34715 - 02/21/2012 12:13 AM - marcandre (Marc-Andre Lafortune)
- proc.c (method_hash, proc_hash): Fix [Unbound]Method#hash
  [Bug #6048]. Isolate hash computation for proc
- internal.h: Declaration for above
- vm_method.c (rb_method_definition_hash): Computation for
  hash part of a method definition
- method.h: Declaration for above
- test/ruby/test_method.rb: Test for above

Revision 34715 - 02/21/2012 12:13 AM - marcandre (Marc-Andre Lafortune)

- proc.c (method_hash, proc_hash): Fix [Unbound]Method#hash
  [Bug #6048]. Isolate hash computation for proc
- internal.h: Declaration for above
- vm_method.c (rb_method_definition_hash): Computation for
  hash part of a method definition
- method.h: Declaration for above
- test/ruby/test_method.rb: Test for above

Revision 2e733be0 - 02/21/2012 07:08 AM - nobu (Nobuyoshi Nakada)

- proc.c (rb_hash_proc): get wrapped pointer properly.  [Bug #6048]
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@34717 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 34717 - 02/21/2012 07:08 AM - nobu (Nobuyoshi Nakada)

- proc.c (rb_hash_proc): get wrapped pointer properly.  [Bug #6048]

Revision 34717 - 02/21/2012 07:08 AM - nobu (Nobuyoshi Nakada)

- proc.c (rb_hash_proc): get wrapped pointer properly.  [Bug #6048]

Revision 34717 - 02/21/2012 07:08 AM - nobu (Nobuyoshi Nakada)

- proc.c (rb_hash_proc): get wrapped pointer properly.  [Bug #6048]

Revision 34717 - 02/21/2012 07:08 AM - nobu (Nobuyoshi Nakada)

- proc.c (rb_hash_proc): get wrapped pointer properly.  [Bug #6048]

Revision 34717 - 02/21/2012 07:08 AM - nobu (Nobuyoshi Nakada)

- proc.c (rb_hash_proc): get wrapped pointer properly.  [Bug #6048]

Revision 34717 - 02/21/2012 07:08 AM - nobu (Nobuyoshi Nakada)

- proc.c (rb_hash_proc): get wrapped pointer properly.  [Bug #6048]

Revision 34717 - 02/21/2012 07:08 AM - nobu (Nobuyoshi Nakada)

- proc.c (rb_hash_proc): get wrapped pointer properly.  [Bug #6048]

Revision 34717 - 02/21/2012 07:08 AM - nobu (Nobuyoshi Nakada)

- proc.c (rb_hash_proc): get wrapped pointer properly.  [Bug #6048]

History
Seems fine.

One thing.
r

rb_method_definition_t* hasn't been used as function interface, rb_method_entry_t* might be preferable.

Hi,

On Mon, Feb 20, 2012 at 3:24 AM, Nobuyoshi Nakada nobu@ruby-lang.org wrote:

Issue #6048 has been updated by Nobuyoshi Nakada.

rb_method_definition_t* hasn't been used as function interface, rb_method_entry_t* might be preferable.

Right, I hesitated about this one. Will change.

Thanks for lookup at my patch.

- Status changed from Open to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r34715.

Marc-Andre, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

- proc.c (method_hash, proc_hash): Fix [Unbound]Method#hash [Bug #6048]. Isolate hash computation for proc
- internal.h: Declaration for above
- vm_method.c (rb_method_definition_hash): Computation for hash part of a method definition
- method.h: Declaration for above
- test/ruby/test_method.rb: Test for above

- Status changed from Closed to Open

This issue was solved with changeset r34717.

Marc-Andre, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

- proc.c (rb_hash_proc): get wrapped pointer properly. [Bug #6048]

Files

| method_hash.patch | 4.32 KB | 02/20/2012 | marcandre (Marc-Andre Lafortune) |