### Ruby master - Bug #16801

The default Struct constructor improperly handle keyword arguments

04/20/2020 09:59 AM - byroot (Jean Boussier)

<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></td>
</tr>
<tr>
<td>Target version:</td>
<td>ruby 2.7.1p83</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.5: DONTNEED, 2.6: DONTNEED, 2.7: REQUIRED</td>
</tr>
</tbody>
</table>

### Description

**Reproduction script:**

```ruby
Field = Struct.new(:value) do
  def initialize(value, keyword: false)
    super(value)
    @keyword = keyword
  end
end

Field.new(1, keyword: true)
```

/tmp/kw.rb:8: warning: Using the last argument as keyword parameters is deprecated; maybe ** should be added to the call
/tmp/kw.rb:2: warning: The called method `initialize' is defined here

This can be worked around with keyword_init: true, but I see no reason why Struct couldn't properly handle keyword arguments in it's default constructor.

I have a patch for this: [https://github.com/ruby/ruby/pull/3045](https://github.com/ruby/ruby/pull/3045) however I had to modify a test that was explicitly expecting that warning, so maybe I'm missing something.

### Associated revisions

Revision adf709a7 - 05/08/2020 08:18 AM - Marc-Andre Laforraine

Classes made from Struct should have default new singleton method.

[Bug #16465] [Bug #16801] [Fix GH-2795] [Fix GH-2944] [Fix GH-3045] [Fix GH-3093]

Note: Backporting shouldn't modify object.h and instead can use struct_new_kw which is basically a duplicate implementation of rb_class_new_instance_pass_kw

Co-authored-by: Yusuke Endoh mame@ruby-lang.org
Co-authored-by: John Hawthorn john@hawthorn.email
Co-authored-by: Adam Hess HParker@github.com
Co-authored-by: Jose Cortinas jacortinas@gmail.com
Co-authored-by: Jean Boussier jean.boussier@gmail.com

### History

**#1 - 04/20/2020 02:05 PM - jeremyevans0 (Jeremy Evans)**

- Backport deleted (2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN)
- ruby -v deleted (ruby 2.7.1p83)
- Tracker changed from Bug to Feature

I think the something your missing is this would break existing code:

```ruby
Struct.new(:hash_value).new(hash_key: 1)
```

Treating the keywords as named columns when they were originally designed to be a hash value for the next column will definitely break existing code. That is why keyword_init must be specified explicitly.
The current design is expected and not a bug. Switching this to a feature request.

#2 - 04/20/2020 02:16 PM - byroot (Jean Boussier)

JeremyEvans0 (Jeremy Evans) I'm afraid I don't understand what you are referring to:

2.7.1:

```ruby
$ ruby --version
ruby 2.7.1p83 (2020-03-31 revision a0c7c23c9c) [x86_64-darwin19]
$ ruby --disable-gems -e 'p Struct.new(:hash_value).new(hash_key: 1)'
#<struct hash_value={:hash_key=>1}>
```

My patch:

```ruby
$ ./ruby --version
ruby 2.8.0dev (2020-04-20T09:16:19Z struct-new-keyword 08d8195ff8) [x86_64-darwin19]
$ ./ruby --disable-gems -e 'p Struct.new(:hash_value).new(hash_key: 1)'
#<struct hash_value={:hash_key=>1}>
```

#3 - 04/20/2020 02:20 PM - byroot (Jean Boussier)

To clarify, what I'm trying to say is that the "new" method should simply forward everything to the initialize method no matter what. I can't see any reason for Struct#new to have arbitrary restrictions.

#4 - 04/20/2020 02:56 PM - JeremyEvans0 (Jeremy Evans)

- Backport set to 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN
- ruby -v set to ruby 2.7.1p83
- Tracker changed from Feature to Bug

JeremyEvans0 (Jeremy Evans) wrote in #note-3:

To clarify, what I'm trying to say is that the "new" method should simply forward everything to the initialize method no matter what. I can't see any reason for Struct#new to have arbitrary restrictions.

Sorry about misunderstanding. Your clarification makes a lot more sense. It seems reasonable that keyword_init would only affect #initialize and not .new.

#5 - 04/21/2020 03:10 AM - nobu (Nobuyoshi Nakada)

- Backport changed from 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN to 2.5: DONTNEED, 2.6: DONTNEED, 2.7: REQUIRED

#6 - 05/08/2020 08:19 AM - Anonymous

- Status changed from Open to Closed

Applied in changeset git|adf709a78534c1483ba8510cb0490464ca31503c.

Classes made from Struct should have default singleton method.

[Bug #16465] [Bug #16801]
[Fix GH-2795] [Fix GH-2944] [Fix GH-3045] [Fix GH-3093]

Note: Backporting shouldn't modify object.h and instead can use
struct_new_kw which is basically a duplicate implementation of
rb_class_new_instance_pass_kw

Co-authored-by: Yuseke Endoh mame@ruby-lang.org
Co-authored-by: John Hawthorn jhn@hawthorn.email
Co-authored-by: Adam Hess HParker@github.com
Co-authored-by: Jose Cortinas jacortinas@gmail.com
Co-authored-by: Jean Boussier jean.boussier@gmail.com