

Ruby trunk - Bug #4440

odd evaluation order in a multiple assignment

02/24/2011 11:05 PM - mame (Yusuke Endoh)

Status: Closed	
Priority: Normal	
Assignee: matz (Yukihiro Matsumoto)	
Target version:	
ruby -v: -	Backport:
Description	
<pre>=begin ##### Ruby ##### def foo p :foo [] end def bar p :bar end x, foo[0] = bar, 0 bar [] foo #####:foo []:bar ##### :bar []:foo ##### ##### obj, obj.foo = obj.foo, obj [] swap ##### splay tree #####: t.left, t.left.right, t = t.left.right, t, t.left 1.9 #####2.0 ##### IRC ##### foo[0] = bar ##### :foo []:bar ##### -- Yusuke Endoh mame@tsg.ne.jp =end</pre>	
Related issues:	
Has duplicate Ruby trunk - Bug #4442: odd evaluation order in a multiple assi...	Rejected 02/24/2011
Has duplicate Ruby trunk - Bug #4443: odd evaluation order in a multiple assi...	Assigned

History

#1 - 02/25/2011 12:23 AM - matz (Yukihiro Matsumoto)

- ruby -v changed from ruby 1.9.2p0 (2010-08-18 revision 29036) [i686-linux] to -

```
=begin
#####
```

In message "Re: [ruby-core:35367] [Ruby 1.9 - Bug #4440] [Open] odd evaluation order in a multiple assignment" on Thu, 24 Feb 2011 23:02:38 +0900, Yusuke Endoh mame@tsg.ne.jp writes:

```
|Ruby
|bar  foo :foo :bar :bar :foo
|1.9
|
|1.8
|
=end
```

#2 - 02/25/2011 12:23 AM - matz (Yukihiko Matsumoto)

```
=begin
Hi,

Oops, I sent Japanese message to the core. Sorry.
```

matz.

In message "Re: [ruby-core:35370] Re: [Ruby 1.9 - Bug #4440] [Open] odd evaluation order in a multiple assignment" on Thu, 24 Feb 2011 23:26:13 +0900, Yukihiko Matsumoto matz@ruby-lang.org writes:

```
|
|In message "Re: [ruby-core:35367] [Ruby 1.9 - Bug #4440] [Open] odd evaluation order in a multiple assignment" on Thu, 24 Feb 2011 23:02:38 +0900, Yusuke Endoh mame@tsg.ne.jp writes:
|
||Ruby
||bar  foo :foo :bar :bar :foo
||1.9
||
||1.8
||
|=end
```

#3 - 02/25/2011 12:23 AM - mame (Yusuke Endoh)

```
=begin
Sorry, I've sent a Japanese mail to ruby-core by mistake.
Translation :-)
```

```
Hi,

I have believed that Ruby evaluates everything from left to right, but was betrayed by a multiple assignment:
```

```
def foo
  p :foo
end
def bar
  p :bar
end

x, foo[0] = bar, 0
```

Because "foo" precedes "bar", I expect that this prints :foo and :bar in this order, but actually :bar and :foo.

I think that this is a matter in practice. For example,

```
x, y = y, x

obj, obj.foo = obj.foo, obj
```

does not so. This kind of code is often written to rotate a tree. In fact I realized this behavior when I was writing

a splay tree, like this:

```
t.left, t.left.right, t = t.left.right, t, t.left
```

I don't think that this should be fixed in 1.9 series, but can we fix this in 2.0?

Some people (in Japanese IRC) said me that the behavior is expected, but note that

```
foo[0] = bar
```

does print :foo and :bar in the order.

--

Yusuke Endoh mame@tsg.ne.jp
=end

#4 - 03/26/2011 10:25 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

#5 - 10/18/2011 09:16 AM - naruse (Yui NARUSE)

- Project changed from Ruby trunk to CommonRuby

- Target version deleted (Next Major)

#6 - 10/23/2011 05:21 PM - naruse (Yui NARUSE)

- Project changed from CommonRuby to Ruby trunk

#7 - 07/14/2012 02:56 PM - ko1 (Koichi Sasada)

- Description updated

- Status changed from Assigned to Closed

duplicate (<http://bugs.ruby-lang.org/issues/4443>)