Ruby master - Bug #5077

method_missing throws NoMemoryError after inheriting from BasicObject

07/22/2011 09:24 PM - madtrick (Farruco Sanjurjo)

Status:	Rejected		
Priority:	Normal		
Assignee:			
Target version:	1.9.2		
ruby -v:	ruby 1.9.2p290 (2011-07-09 revision 32478) [x86_64-linux]	**Backport:**	

Description

If a class inherits from BasicObject and then overwrites method_missing like this:

class A < BasicObject
def method_missing(*a)
puts "#{a}"
end
end

And we try it:

A.new.fooooo

The interpreter enters what looks like a loop and then crashes with this trace (in irb):

NoMemoryError: failed to allocate memory
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'
from (irb):3:in method_missing'
from (irb):3:inmethod_missing'

If we create the same class with the same method_missing but without inheriting from BasicObject it works right.

History

#1 - 07/22/2011 09:58 PM - Eregon (Benoit Daloze)

Farruco Sanjurjo wrote:

> If a class inherits from BasicObject and then overwrites method_missing like this:
>
> class A < BasicObject
> def method_missing(*a)
> puts "#{a}"
> end
> end
>
> And we try it:
>
> A.new.fooooo
>
> The interpreter enters what looks like a loop and then crashes with this trace (in irb):

method_missing is called indefinitely recursively because "puts" is not a method in BasicObject.

Only methods available in BasicObject are:
BasicObject.instance_methods # => [:==, :equal?, :!, :!=, :instance_eval, :instance_exec, :**send**]

"puts" is usually provided by Kernel#puts. But Kernel is not included in BasicObject.
You could solve this two ways:

- use "::Kernel::puts" instead of "puts" (the documentation about the first :: is being discussed in Bug #5067)
- include Kernel into your class A

#2 - 07/22/2011 10:19 PM - mrkn (Kenta Murata)

- Status changed from Open to Rejected

- Priority changed from 5 to Normal

BasicObject doesn't include Kernel module.
It is a spec.