Ruby trunk - Bug #13498

Weakref, Weakmap and define_finalizer don't work on frozen objects

04/23/2017 07:16 AM - herwin (Herwin W)

Status:	Assigned		
Priority:	Normal		
Assignee:	nobu (Nobuyoshi Nakada)		
Target version:			
ruby -v:	ruby 2.4.1p111 (2017-03-22 revision 58053) [x86_64-linux] (but seen the same issue with 2.3 and 2.1)	**Backport:**	2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description

I'm just creating a single ticket for these issues, I guess they're actually all the same (I've seen weakref uses weakmap, not sure about define_finalizer).

```
require 'weakref'

map = ObjectSpace::WeakMap.new

o = Object.new
o.freeze

begin
  WeakRef.new(o)
rescue => e
  STDERR.puts e
end

begin
  map[o] = 'foo'
rescue => e
  STDERR.puts e
end

begin
  map['bar'] = o
rescue => e
  STDERR.puts e
end

begin
  ObjectSpace.define_finalizer(o, ->(id) { p id })
rescue => e
  STDERR.puts e
end
```

Every statement here raises the runtime error "can't modify frozen Object". The documentation doesn't mention that frozen objects are not allowed, the closest reference we get is a short paragraph in WeakRef: "With this you will have to limit your self to String keys, otherwise you will get an ArgumentError because WeakRef cannot create a finalizer for a Symbol. Symbols are immutable and cannot be garbage collected"

History

#1 - 06/16/2017 02:36 AM - nobu (Nobuyoshi Nakada)

- Assignee set to nobu (Nobuyoshi Nakada)

- Status changed from Open to Assigned

Weakref is implemented by finalizer to notify that an object is collected, and you can't define finalizers on frozen objects.
Probably we need to move finalizer flags to a separate region (like bitmap marking).

#2 - 10/24/2017 06:53 PM - RubyBugs (A Nonymous)

nobu (Nobuyoshi Nakada) wrote:

> Weakref is implemented by finalizer to notify that an object is collected, and you can't define finalizers on frozen objects.
> Probably we need to move finalizer flags to a separate region (like bitmap marking).

Yes please. I also came here having discovered this independently, and also see it as a bug. It makes designs using frozen objects difficult to this friction with a leaky abstraction in the language implementation.

Unless I am mistaken, and mutating objects is an intentional part of the design of the GC system?