Ruby - Feature #9834

Float#{next_float,prev_float}

05/13/2014 09:48 AM - akr (Akira Tanaka)

Status:	Closed
Priority:	Normal
Assignee:	akr (Akira Tanaka)
Target version:	

Description

I'd like to add Float#next_float and Float#prev_float which
returns next representable floating-point number and
previous representable floating-point number.

```
p 3.0.next_float #=> 3.0000000000000004
p 3.0.prev_float #=> 2.9999999999999996
```

These methods can be useful to examine the behavior of floating-point numbers.

For example, they can be used to examine floating-point error in 0.1 + 0.1 + ... + 0.1.

```
f = 0.0
100.times { f += 0.1 }
p f                           #=> 9.99999999999998       # should be 10.0 in the ideal world.
p 10-f                        #=> 1.9539925233402755e-14 # the floating-point error.
p(10.0.next_float-10)         #=> 1.7763568394002505e-15 # 1 ulp (units in the last place).
p((10-f)/(10.0.next_float-10)) #=> 11.0                  # the error is 11 ulp.
p "%a" % f                    #=> "0x1.3fffffffffff5p+3" # the last hex digit is 5.  16 - 5 = 11 ulp.
```

The methods are implemented using nextafter() function described in
IEEE 754 (Appendix), C99 and POSIX.
It seems the function is pretty portable on Unix variants.

However I implemented missing/nextafter.c for environments which don't have the function.

Any idea?

Associated revisions

Revision 63fee735002b34d37598d4830ef35073202dda58 - 05/18/2014 12:37 AM - akr (Akira Tanaka)

- configure.in: Check nextafter() availability.
- include/ruby/missing.h (nextafter): New optional declaration.
- missing/nextafter.c: New file.
- numeric.c: Float#next_float and Float#prev_float implemented.

  [ruby-core:62562] [Feature #9834]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45982 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 63fee735 - 05/18/2014 12:37 AM - akr (Akira Tanaka)

- configure.in: Check nextafter() availability.
- include/ruby/missing.h (nextafter): New optional declaration.

- missing/nextafter.c: New file.
- numeric.c: Float#next_float and Float#prev_float implemented.

  [ruby-core:62562] [Feature #9834]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45982 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision d7df70c08f25f4f21d0c5d7690657faae92026b0 - 05/18/2014 05:46 AM - nobu (Nobuyoshi Nakada)

Makefile.sub: fix nextafter

- win32/Makefile.sub (MISSING, CONFIG_H): msvcr120.dll provides
  nextafter() as well as other mathematic functions. finite() and
  isnan() used in missing/nextafter.c are not by older runtimes.
  [Feature #9834]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45992 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision d7df70c0 - 05/18/2014 05:46 AM - nobu (Nobuyoshi Nakada)

Makefile.sub: fix nextafter

- win32/Makefile.sub (MISSING, CONFIG_H): msvcr120.dll provides
  nextafter() as well as other mathematic functions. finite() and
  isnan() used in missing/nextafter.c are not by older runtimes.
  [Feature #9834]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45992 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 05/13/2014 10:28 AM - phasis68 (Heesob Park)

Here is a pure ruby implementation of Float#{next_float,prev_float} (adopted from http://golang.org/src/pkg/math/nextafter.go)

```
class Float
  def dbl2num(dbl)
    [dbl].pack('d').unpack('Q')[0]
  end

  def num2dbl(num)
    [num].pack('Q').unpack('d')[0]
  end

  def nextafter(y)
    y = y.to_f
    if (self.nan? || y.nan?)
      Float::NAN
    elsif self == y
      y
    elsif self == 0
      num2dbl(1) * (y<=>0.0)
    elsif (y > self) == (self > 0)
      num2dbl(dbl2num(self) + 1)
    else
      num2dbl(dbl2num(self) - 1)
    end
  end

  def prev_float
    nextafter(-Float::INFINITY)
  end

  def next_float
    nextafter(Float::INFINITY)
  end
end
```

#2 - 05/13/2014 12:27 PM - akr (Akira Tanaka)

Thank you for an interesting implementation.

Heesob Park wrote:

```
      if (self==Float::NAN || y==Float::NAN)
```

This doesn't work. Float#nan? should be used.

```
      elsif self == y
        r = self
```

This should be "r = y" to to follow C99's nextafter() behavior.
(It doesn't affect next_float and prev_float, though.)

#3 - 05/17/2014 07:30 AM - matz (Yukihiro Matsumoto)

Accepted.

Matz.

#4 - 05/17/2014 12:17 PM - marcandre (Marc-Andre Lafortune)

Float#next or Float#next_float?

Yukihiro Matsumoto wrote:

> Accepted.
>
> Matz.

#5 - 05/17/2014 12:48 PM - matz (Yukihiro Matsumoto)

- Assignee set to akr (Akira Tanaka)

Float#next_float definitely. Float#next is too short and too simple for this method.

Matz.

#6 - 05/18/2014 12:37 AM - akr (Akira Tanaka)

- Status changed from Open to Closed

- % Done changed from 0 to 100

Applied in changeset r45982.

- configure.in: Check nextafter() availability.
- include/ruby/missing.h (nextafter): New optional declaration.
- missing/nextafter.c: New file.
- numeric.c: Float#next_float and Float#prev_float implemented.

  [ruby-core:62562] [Feature #9834]

Files

next_float-and-prev_float.patch	11.9 KB	05/13/2014	akr (Akira Tanaka)