

Ruby master - Bug #13078

with Integer a and b, a.fdiv(b) sometimes inaccurate than Rational(a, b).to_f

12/27/2016 06:36 AM - metanest (Makoto Kishimoto)

Status: Closed	
Priority: Normal	
Assignee:	
Target version:	
ruby -v:	Backport: 2.2: REQUIRED, 2.3: REQUIRED, 2.4: DONE
Description For example, <pre>a = 3 b = 0x20_0000_0000_0001 p a.fdiv(b) p Rational(a, b).to_f #=> 3.3306690738754696e-16 3.330669073875469e-16</pre> <p>I'm hacking to solve this problem, see diff view. https://github.com/ruby/ruby/compare/trunk...metanest:fdiv_spike</p>	

Associated revisions

Revision 1778ed59 - 12/28/2016 04:40 AM - nobu (Nobuyoshi Nakada)

numeric.c: reduce fdiv

- numeric.c (rb_int_fdiv_double): reduce first for more precise result. [ruby-core:78886] [Bug #13078]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@57227 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 57227 - 12/28/2016 04:40 AM - nobu (Nobuyoshi Nakada)

numeric.c: reduce fdiv

- numeric.c (rb_int_fdiv_double): reduce first for more precise result. [ruby-core:78886] [Bug #13078]

Revision 57227 - 12/28/2016 04:40 AM - nobu (Nobuyoshi Nakada)

numeric.c: reduce fdiv

- numeric.c (rb_int_fdiv_double): reduce first for more precise result. [ruby-core:78886] [Bug #13078]

Revision 57227 - 12/28/2016 04:40 AM - nobu (Nobuyoshi Nakada)

numeric.c: reduce fdiv

- numeric.c (rb_int_fdiv_double): reduce first for more precise result. [ruby-core:78886] [Bug #13078]

Revision ea52cac1 - 03/11/2017 01:03 PM - naruse (Yui NARUSE)

merge revision(s) 57227: [Backport #13078]

```
numeric.c: reduce fdiv
```

```
* numeric.c (rb_int_fdiv_double): reduce first for more precise
result. [ruby-core:78886] [Bug #13078]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_4@57840 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 57840 - 03/11/2017 01:03 PM - naruse (Yui NARUSE)

merge revision(s) 57227: [Backport #13078]

numeric.c: reduce fdiv

```
* numeric.c (rb_int_fdiv_double): reduce first for more precise
result. [ruby-core:78886] [Bug #13078]
```

History

#1 - 12/28/2016 03:00 AM - wanabe (_ wanabe)

The accuracy of Rational(a, b).to_f seems to be due to reduction of fraction.
How about using Rational object, or its internal function f_gcd() and f_idiv()?

#2 - 12/28/2016 03:11 AM - nobu (Nobuyoshi Nakada)

I thought the reduction would be slower, but actually not significant.

```
$ ruby -rbenchmark -e 'a = 3; b = -0x20_0000_0000_0001; n=10_000_000; Benchmark.bm(13) {|x|x.report("Integer#f
div"){n.times{a.fdiv(b)}}; x.report("Rational#to_f"){n.times{Rational(a, b).to_f}}}'
      user      system      total      real
Integer#fdiv  3.420000    0.000000    3.420000 ( 3.446264)
Rational#to_f 3.450000    0.010000    3.460000 ( 3.465592)
```

#3 - 12/28/2016 04:41 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

Applied in changeset r57227.

numeric.c: reduce fdiv

- numeric.c (rb_int_fdiv_double): reduce first for more precise result. [ruby-core:78886] [Bug #13078]

#4 - 12/30/2016 12:53 PM - metanest (Makoto Kishimoto)

pull requested

<https://github.com/ruby/ruby/pull/1508>

#5 - 03/11/2017 01:03 PM - naruse (Yui NARUSE)

- Backport changed from 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN to 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: DONE

ruby_2_4 r57840 merged revision(s) 57227.

#6 - 03/13/2017 02:30 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: DONE to 2.2: UNKNOWN, 2.3: REQUIRED, 2.4: DONE

#7 - 03/23/2017 03:26 AM - usa (Usaku NAKAMURA)

- Backport changed from 2.2: UNKNOWN, 2.3: REQUIRED, 2.4: DONE to 2.2: REQUIRED, 2.3: REQUIRED, 2.4: DONE