Ruby - Feature #14042

IO#puts: use writev if available

10/22/2017 03:41 PM - rohitpaulk (Paul Kuruvilla)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		

Description

Hi,

I've attached a patch to make IO#puts use writev if available. Currently, IO#puts calls write twice: Once to write the string, and the second to write a newline (if the string doesn't end with one already). With this patch, those two calls are replaced with a single writev call.

A test has been added that demonstrates the problem.

For a bit of background:

- A related issue: https://bugs.ruby-lang.org/issues/9420. (I couldn't figure out a way to 'attach' my patch to that issue, so I'm creating a new one)
- A blog post I wrote a while back about this: https://hackernoon.com/rubys-puts-is-not-atomic-889c57fc9a28

Command I used to run the test I added: make test-all TESTS='ruby/test_io.rb -n test_puts_parallel'

I'm a first time contributor, a bit confused as to where a changelog entry should be added. Is the NEWS file the right place?

Regards,
Paul

Related issues:

Related to Ruby - Feature #9323: IO#writev	**Closed**
Is duplicate of Ruby - Feature #9420: warn and puts should be atomic	**Closed**

Associated revisions

Revision 635d0822cdb3a9eb7612ea478601d17ebe76b74d - 10/25/2017 05:44 AM - nobu (Nobuyoshi Nakada)

io.c: write a newline together

- io.c (rb_io_puts): write a newline together at once for each argument. based on the patch by rohitpaulk (Rohit Kuruvilla) at [ruby-core:83508]. [Feature #14042]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60417 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 635d0822 - 10/25/2017 05:44 AM - nobu (Nobuyoshi Nakada)

io.c: write a newline together

- io.c (rb_io_puts): write a newline together at once for each argument. based on the patch by rohitpaulk (Rohit Kuruvilla) at [ruby-core:83508]. [Feature #14042]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60417 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 237901a41b44dd86754b0a02979d03b56faeeb73 - 10/25/2017 12:04 PM - nobu (Nobuyoshi Nakada)

io.c: warn old write

- io.c (rb_io_puts): warn if write method accepts just one argument. [ruby-core:83529] [Feature #14042]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60423 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 237901a4 - 10/25/2017 12:04 PM - nobu (Nobuyoshi Nakada)

io.c: warn old write

- io.c (rb_io_puts): warn if write method accepts just one
  argument. [ruby-core:83529] [Feature #14042]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60423 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 7da5b32a7df809783726a9aa2d37ed56822f91c7 - 10/25/2017 12:31 PM - nobu (Nobuyoshi Nakada)

test_io.rb: skip writev test

- test/ruby/test_io.rb (TestIO#test_puts_parallel): skip a test
  needs writev which is not portable. [Feature #14042]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60424 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 7da5b32a - 10/25/2017 12:31 PM - nobu (Nobuyoshi Nakada)

test_io.rb: skip writev test

- test/ruby/test_io.rb (TestIO#test_puts_parallel): skip a test
  needs writev which is not portable. [Feature #14042]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60424 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 8354b6d2cd0b83ad801d038a034e78e397ca32db - 10/29/2017 05:46 AM - nobu (Nobuyoshi Nakada)

io.c: honor buffered mode

- io.c (io_writev): honor buffered mode to get rid of broken pipe
  error when stdout is redirected to a pipeline.
  [ruby-core:83578] [Feature #14042]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60535 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 8354b6d2 - 10/29/2017 05:46 AM - nobu (Nobuyoshi Nakada)

io.c: honor buffered mode

- io.c (io_writev): honor buffered mode to get rid of broken pipe
  error when stdout is redirected to a pipeline.
  [ruby-core:83578] [Feature #14042]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60535 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 10/23/2017 04:36 AM - shyouhei (Shyouhei Urabe)

- Related to Feature #9323: IO#writev added

#2 - 10/23/2017 08:05 AM - nobu (Nobuyoshi Nakada)

- Is duplicate of Feature #9420: warn and puts should be atomic added

#3 - 10/23/2017 08:20 AM - nobu (Nobuyoshi Nakada)

It breaks other tests which mock $stdout and $stderr.

#4 - 10/23/2017 08:30 AM - shyouhei (Shyouhei Urabe)

nobu (Nobuyoshi Nakada) wrote:

> It breaks other tests which mock $stdout and $stderr.

I strongly believe implementation details of IO#puts shall never be what a test should care about. I think it's okay to suppress them. @rohitpaulk can you also provide changes to our tests so that make check should pass?

#5 - 10/23/2017 11:03 AM - Eregon (Benoit Daloze)

shyouhei (Shyouhei Urabe) wrote:

> I strongly believe implementation details of IO#puts shall never be what a test should care about.

I agree.
For information, ruby/spec used to check write calls from puts (~1 year ago) but I changed it to only care about what's written, not individual calls.
FWIW, TruffleRuby has an atomic #puts, currently simply by adding \n to the String before the write() call, if it does not already end with it.

#6 - 10/23/2017 11:57 AM - mame (Yusuke Endoh)

shyouhei (Shyouhei Urabe) wrote:

> nobu (Nobuyoshi Nakada) wrote:
>
> > It breaks other tests which mock $stdout and $stderr.
>
> I strongly believe implementation details of IO#puts shall never be what a test should care about.

I'm unsure if it is a spec or not, but at least, it is a well-known practice to assign to $stdout a dummy object that implements a write method:

- https://sketchucation.com/forums/viewtopic.php?f=180&t=31160
- http://d.hatena.ne.jp/rubikitch/20080421/1208758284

Matz has also showed a code using the practice:

http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-list/42076

I think we need to care the existing programs that uses the practice.

#7 - 10/23/2017 12:07 PM - rohitpaulk (Paul Kuruvilla)

- File ruby-changes.patch added

Patch updated (attached) to fix the failures mentioned.

There were 10 failures, all within test/mkmf. The mocked IO object's #write method has been changed to support the interface change in https://bugs.ruby-lang.org/projects/ruby-trunk/repository/revisions/60343

```
-    def write(s)
+    def write(*args)
       if @out
-        @buffer << s
+        @buffer << args.join
       elsif @origin
-        @origin << s
+        @origin << args.join
       end
     end
```

#8 - 10/23/2017 12:16 PM - Eregon (Benoit Daloze)

mame (Yusuke Endoh) wrote:

> I'm unsure if it is a spec or not, but at least, it is a well-known practice to assign to $stdout a dummy object that implements a write method:

Good catch!
The patch calls #write, but with 2 arguments.
Probably one needs to check that the receiver #write can accept multiple arguments,
or do this logic only is $stdout.write is the original IO#write.

#9 - 10/23/2017 12:21 PM - rohitpaulk (Paul Kuruvilla)

- File deleted (ruby-changes.patch)

#10 - 10/23/2017 12:24 PM - rohitpaulk (Paul Kuruvilla)

> Probably one needs to check that the receiver #write can accept multiple arguments

I think that'd be a good approach to ensure that mocked objects in existing programs don't break.

Would it make sense to add a deprecation warning in this case? i.e. If the receiver doesn't accept multiple arguments, we emit a deprecation warning and make 2 calls instead of one. If the receiver accepts multiple arguments, we make a single call.

#11 - 10/23/2017 01:13 PM - rohitpaulk (Paul Kuruvilla)

> Would it make sense to add a deprecation warning in this case? i.e. If the receiver doesn't accept multiple arguments, we emit a deprecation warning and make 2 calls instead of one. If the receiver accepts multiple arguments, we make a single call.

Something along the lines of...

```
 rb_io_writev(VALUE io, int argc, VALUE *argv)
 {
-    return rb_funcallv(io, id_write, argc, argv);
+    if (rb_obj_method_arity(io, id_write) == -1) {
+        rb_funcallv(io, id_write, argc, argv);
+    }
+    else {
+        /**
+         * Previously, IO#write only accepted one argument. This was changed
+         * to use multiple arguments in revision #60343.
+         *
+         * To play well with programs that might've mocked an IO object and are
+         * only expecting a single argument - let's make multiple calls with
+         * a single argument each.
+         */
+        rb_warn("IO#write has been changed to accept multiple arguments. \
+You are seeing this warning because an object expected to implement the IO \
+interface has a #write method that doesn't accept multiple arguments. Please \
+change the implementation to accept multiple arguments.");
+        for (int i = 0; i < argc; i++) {
+            rb_io_write(io, argv[i]);
+        }
+    }
 }
```

```
+  def test_puts_works_with_io_objects_that_only_accept_single_arg
+    klass = Class.new do
+      attr_reader :captured
+
+      def write(str)
+        (@captured ||= "") << str
+      end
+    end
+
+    mocked_io_obj = klass.new
+    old_stdout, $stdout = $stdout, mocked_io_obj
+    puts "hey" # Should write to the mocked class
+    assert_equal("hey\n", mocked_io_obj.captured)
+  ensure
+    $stdout = old_stdout
+  end
+
```

#12 - 10/25/2017 05:44 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

Applied in changeset trunk|r60417.

---

io.c: write a newline together

- io.c (rb_io_puts): write a newline together at once for each
  argument. based on the patch by rohitpaulk (Rohit Kuruvilla) at
  [ruby-core:83508]. [Feature #14042]

#13 - 10/26/2017 12:24 PM - Eregon (Benoit Daloze)

I think it would be safer to only do this if we know we are calling the original IO#write.
Seeing r60428 and how it breaks mspec too, the current approach sounds not compatible enough.

For instance, if write is defined as write(*args), there is no way to know whether that the arguments are just passed to IO#write or does some of its own parsing, which will very likely break with this change.
As an example (not realistic as only keeps the last call to #write),

```
require 'pathname'
$stdout = Pathname.new("log")
puts "foo"
```

raises an error with this change:

```
Traceback (most recent call last):
4: from test.rb:3:in `<main>'
3: from test.rb:3:in `puts'
2: from test.rb:3:in `puts'
1: from test.rb:3:in `write'
test.rb:3:in `write': no implicit conversion of String into Integer (TypeError)
```

#14 - 10/26/2017 12:41 PM - Eregon (Benoit Daloze)

Correction, it only breaks one spec, not mspec.
The spec is "IO.popen raises IOError when writing a read-only pipe" in popen_spec.rb.

I reduced the code and this behaves differently:

```
$ chruby trunk
$ ruby -e 'IO.popen(["ruby", "-e", "sleep 1; puts :a"], "r").close'
Traceback (most recent call last):
    3: from -e:1:in `<main>'
    2: from -e:1:in `puts'
    1: from -e:1:in `puts'
-e:1:in `write': Broken pipe @ io_writev - <STDOUT> (Errno::EPIPE)

$ chruby 2.4.2
$ ruby -e 'IO.popen(["ruby", "-e", "sleep 1; puts :a"], "r").close'
```

I don't understand why though.

#15 - 10/29/2017 04:34 PM - Eregon (Benoit Daloze)

Nobu told me it is because of different buffering with writev, i.e. no buffering.
Nobu fixed it in r60535.
The spec was also racy (child #write+flush raced with parent #close), so I fixed it in r60567.

Files

ruby-changes.patch	2.83 KB	10/23/2017	rohitpaulk (Paul Kuruvilla)