Lazy enumerators should want to stay lazy.

01/19/2013 02:33 AM - marcandre (Marc-Andre Lafortune)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>marcandre</td>
</tr>
<tr>
<td>Target version</td>
<td>2.0.0</td>
</tr>
<tr>
<td>ruby -v:</td>
<td>r38825</td>
</tr>
</tbody>
</table>

Description

I'm just waking up to the fact that many methods turn a lazy enumerator in a non-lazy one.

Here's an example from Benoit Daloze in [ruby-core:44151]:

```ruby
lines = File.foreach('a_very_large_file').lazy
  .select {|line| line.length < 10 }
  .map {|line| line.chomp!; line }
  .each_slice(3)
  .map {|lines| lines.join(';').downcase }
  .take_while {|line| line.length > 20 }
```

That code will produce the right result but will read the whole file, which is not what is desired.

Indeed, each_slice currently does not return a lazy enumerator :-(

To make the above code as intended, one must call .lazy right after the each_slice(3). I feel this is dangerous and counter intuitive.

Is there a valid reason for this behavior? Otherwise, I would like us to consider returning a lazy enumerator for the following methods:

- (when called without a block)
  - each_with_object
  - each_with_index
  - each_slice
  - each_entry
  - each_cons
  - (always)
  - chunk
  - slice_before

The arguments are:

- fail early (much easier to realize one needs to call a final force, to_a or each than realizing that a lazy enumerator chain isn't actually lazy)
- easier to remember (every method normally returning an enumerator returns a lazy enumerator). basically this makes Lazy covariant
- I'd expect that if you get lazy at some point, you typically want to remain lazy until the very end

Associated revisions

Revision 1677cbce - 02/05/2013 03:49 AM - marcandre (Marc-Andre Lafortune)

- enumerator.c: Use to_enum for Enumerable methods returning Enumerators.
  This makes Lazy#cycle no longer needed, so it was removed.
  Make Enumerator#chunk and slice_before return lazy Enumerators.
  [Bug #7715]
- internal.h: Remove ref to rb_enum_cycle_size; no longer needed
- enum.c: Make enum_cycle_size static.
- test/ruby/test_lazy Enumerator.rb: Test for above
enum.c: Use to_enum for Enumerable methods returning Enumerators. This makes Lazy#cycle no longer needed, so it was removed. Make Enumerator#chunk and slice_before return lazy Enumerators. [Bug #7715]

internal.h: Remove ref to rb_enum_cycle_size; no longer needed

e num.c: Make enum_cycle_size static.

test/ruby/test_lazy_enumerator.rb: Test for above

Revision 39058 - 02/05/2013 03:49 AM - marcandre (Marc-Andre Lafortune)

enum.c: Use to_enum for Enumerable methods returning Enumerators. This makes Lazy#cycle no longer needed, so it was removed. Make Enumerator#chunk and slice_before return lazy Enumerators. [Bug #7715]

internal.h: Remove ref to rb_enum_cycle_size; no longer needed

e num.c: Make enum_cycle_size static.

test/ruby/test_lazy_enumerator.rb: Test for above

Revision 39058 - 02/05/2013 03:49 AM - marcandre (Marc-Andre Lafortune)

enum.c: Use to_enum for Enumerable methods returning Enumerators. This makes Lazy#cycle no longer needed, so it was removed. Make Enumerator#chunk and slice_before return lazy Enumerators. [Bug #7715]

internal.h: Remove ref to rb_enum_cycle_size; no longer needed

e num.c: Make enum_cycle_size static.

test/ruby/test_lazy_enumerator.rb: Test for above

Revision 39058 - 02/05/2013 03:49 AM - marcandre (Marc-Andre Lafortune)

enum.c: Use to_enum for Enumerable methods returning Enumerators. This makes Lazy#cycle no longer needed, so it was removed. Make Enumerator#chunk and slice_before return lazy Enumerators. [Bug #7715]

internal.h: Remove ref to rb_enum_cycle_size; no longer needed

e num.c: Make enum_cycle_size static.

test/ruby/test_lazy_enumerator.rb: Test for above
enumerator.c: Use to_enum for Enumerable methods returning Enumerators. This makes Lazy#cycle no longer needed, so it was removed. Make Enumerator#chunk and slice_before return lazy Enumerators. [Bug #7715]

internal.h: Remove ref to rb_enum_cycle_size; no longer needed
enum.c: Make enum_cycle_size static.
test/ruby/test_lazy Enumerator.rb: Test for above

enumerator.c: Use to_enum for Enumerable methods returning Enumerators. This makes Lazy#cycle no longer needed, so it was removed. Make Enumerator#chunk and slice_before return lazy Enumerators. [Bug #7715]

internal.h: Remove ref to rb_enum_cycle_size; no longer needed
enum.c: Make enum_cycle_size static.
test/ruby/test_lazy Enumerator.rb: Test for above

---

**History**

#1 - 01/25/2013 12:09 PM - ko1 (Koichi Sasada)
- Target version set to 2.0.0

Who's ball?

#2 - 01/25/2013 03:20 PM - marcandre (Marc-Andre Lafoutine)
- Status changed from Open to Assigned
- Assignee set to marcandre (Marc-Andre Lafoutine)

I can do it, unless there are objections.

#3 - 01/25/2013 06:24 PM - shugo (Shugo Maeda)
marcandre (Marc-Andre Lafoutine) wrote:

I can do it, unless there are objections.

Your proposal sounds reasonable.
I guess these methods were forgotten to change when lazy was implemented.

#4 - 01/25/2013 06:29 PM - yhara (Yutaka HARA)
shugo (Shugo Maeda) wrote:

I guess these methods were forgotten to change when lazy was implemented.

That's right -(- I thought these methods does not need to be overridden because they return Enumerator, but they should return Enumerator::Lazy for such cases.

#5 - 02/04/2013 09:12 AM - marcandre (Marc-Andre Lafoutine)
I believe I have found the key to resolve this issue, Lazy.new issue [#7248] and others.
We simply need to specialize to_enum/enum_for for lazy enumerators.

In the same way, RETURN_SIZED_ENUMERATOR should return a lazy enumerator, when called for a lazy enumerator.

With this in mind:

- Lazy.each_with_object, etc..., will correctly return lazy enumerators [#7715] without being overridden.
- Lazy#cycle can be removed. It no longer needs to be overridden.
- Lazy.new really has no need to accept (method, *args) and can be modified as proposed in [#7248]
- Any user method of Enumerable that returns an Enumerator using to_enum will conserve laziness.

None of this could create a regression, since Lazy & RETURN_SIZED_ENUMERATOR are both new to 2.0.0

I'm working on a patch...

#6 - 02/04/2013 04:52 PM - marcandre (Marc-Andre Lafortune)
Patch almost done, which also fixes #7248
https://github.com/marcandre/ruby/compare/marcandre:master...marcandre:lazy

Still missing:

- tweak inspect
- fix .lazy.size
- couple more tests

#7 - 02/05/2013 07:32 AM - marcandre (Marc-Andre Lafortune)
Patch updated, rdoc improved too.

Makes for a clean API for Lazy#new also, and there's even less code (~20 loc).

I'll review the patch one last time before committing it (in about 5 hours).

#8 - 02/05/2013 12:50 PM - marcandre (Marc-Andre Lafortune)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r39058.
Marc-Andre, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- enumerator.c: Use to_enum for Enumerable methods returning Enumerators. This makes Lazy#cycle no longer needed, so it was removed. Make Enumerator#chunk and slice_before return lazy Enumerators. [Bug #7715]
- internal.h: Remove ref to rb_enum_cycle_size; no longer needed
- enum.c: Make enum_cycle_size static.
- test/ruby/test_lazy Enumerator.rb: Test for above