Ruby - Feature #8639

Add Queue#each

07/16/2013 12:44 AM - avdi (Avdi Grimm)

Status:	Closed	
Priority:	Normal	
Assignee:	ko1 (Koichi Sasada)	
Target version:	2.6	

Description

I was fiddling around with Queue the other day and realized it doesn't have an #each method. So I made one:
https://github.com/ruby/ruby/pull/361

As for why, it makes for a convenient way to build consumer processes:

```
inq = Queue.new
outq = Queue.new
doubler = Thread.new do
  inq.each do |n|
    outq << n + n
  end
end
```

My PR also returns an Enumerator when no block is given, and handles the non_block argument.

I'm sure there's some good reason that this method wasn't there already, so feel free to explain.

Also, this is my first Ruby feature ticket so please let me know if I'm missing any points of protocol.

Thanks!

Related issues:

Related to Ruby - Bug #10485: NoMethodError "undefined method `initialize_cop...	**Closed**

History

#1 - 07/16/2013 12:54 AM - rkh (Konstantin Haase)

+1 for the feature, should have gone into CommonRuby, I think, maybe.

#2 - 07/16/2013 01:23 AM - avdi (Avdi Grimm)

On Mon, Jul 15, 2013 at 11:54 AM, rkh (Konstantin Haase) me@rkh.im wrote:

> +1 for the feature, should have gone into CommonRuby, I think.

Is that something I can change?

#3 - 07/16/2013 01:23 AM - avdi (Avdi Grimm)

P.S. Should I reply on the ticket or do these conversations eventually get
posted back to it? I'm new to all of this!

#4 - 07/16/2013 05:29 AM - regularfry (Alex Young)

On Tue, 2013-07-16 at 00:44 +0900, avdi (Avdi Grimm) wrote:

> Issue #8639 has been reported by avdi (Avdi Grimm).

I thought this sounded familiar. It has come up before:
http://bugs.ruby-lang.org/issues/4589

--
Alex

#5 - 07/16/2013 05:53 AM - avdi (Avdi Grimm)

On Mon, Jul 15, 2013 at 4:24 PM, Alex Young alex@blackkettle.org wrote:

> I thought this sounded familiar. It has come up before:
> http://bugs.ruby-lang.org/issues/4589

Hey, thanks for finding that! I figured it had probably come up before.
Some notes responding to that conversation:

- First, as Alex pointed out then, IO#each is often destructive in the sense that it's either moving the read pointer forward or (in the case of a pipe or socket) losing info once it advances. So there IS a precedent.
- I feel like the semantics I've implemented in the PR are moderately sane, but feel free to point out anything I've missed.
- "Loop forever over the things coming out of this queue until something breaks the loop" seems like a common enough use case to warrant some sugar.
- Note this is NOT "loop until the queue is exhausted", unless you turn non_block on.
- I am explicitly not discussing the inclusion of Enumerable in this ticket. That's a much bigger can of worms and I haven't even begun to think through all the implications.
- Note, though, that with this PR if you DO want a full Enumerable over a queue all you have to do now is say q.each, omitting the block. So we're not making Queue enumerable, but we're making it easy to get at a queue Enumerator if you really want one. Without this PR the shortest code I've found to do that is:

```
q = Queue.new
eq = Enumerator.new do |y|
  loop do
    y << q.shift
  end
end
```

...and this code doesn't cover all the cases that I've covered in the PR.

#6 - 07/16/2013 06:23 AM - avdi (Avdi Grimm)

BTW, I feel like I've left out the biggest justification for this, which for me is POLS. What prompted me to submit the PR was that I actually started writing an example of using Queue#each---it just made sense for it to exist, so much so I thought I remembered using it before---and then realized it wasn't there. To me it felt very natural and intuitive to expect Queue#each to exist. YMMV, obviously, but #each is such a widespread convention in Ruby that it seems natural to look for it on a Queue class.

OK, I'll shut up now :-)

#7 - 08/07/2013 09:53 AM - ko1 (Koichi Sasada)

Please assume
@q_buf is current buffer (array) of queue in following example.
@q_lock is lock of this Queue.

People think Queue#each should be:

(1) Infinite loop

```
# iterate forever
def each
  loop{
    yield @q_buf.shift
  }
end
```

(2) Finite loop

(2-1) Mutable behavior

```
# iterate untile elements are exists
def each
  loop{
    e = nil
    @q_lock.lock{
```

```
      return if @q_buf.empty?
      e = @q_buf.shift
    }
    yield e
  }
end
```

(2-2) Immutable behaviour

```
def each(&b)
  @q_buf.dup.each(&b)
end
# only for chekcking current elements
# not for inter-thraed communication
```

Please add another possible version.

Your proposal is (1). (1) is shorter version of

```
while e = q.pop
  break if e == :end
  ...
end
```

which I wrote frequently.

```
q.each{|e|
  break if e == :end
  ...
}
```

I'm weak negative because
(a) seems not so convinient
(b) #each method is for Enumerable
If we find #each method, then we think it is Enumerable.
(c) we have only few cases for infinite iteration #each methods.
(d) I can think 3 versions described above with in seconds.
(e) I want to use Queue with push' and pop' operations.
Operating with `push' and #each is not consistent.
(Of course, this is my thought. not practical reason)

BTW, I think your `non-blocking' parameter should be a switch of (1) and (2-1).

#8 - 08/09/2013 07:42 PM - ko1 (Koichi Sasada)

- Assignee set to ko1 (Koichi Sasada)

#9 - 08/15/2013 06:00 AM - zzak (zzak _)

- File 361.patch added

Attaching the patch from Avdi's pull request on github, thank you!

#10 - 08/15/2013 07:00 AM - avdi (Avdi Grimm)

Thanks! I'm off to a conf this week, it probably would have been a awhile
before I got around to it.

--
Avdi Grimm
http://avdi.org

I only check email twice a day. to reach me sooner, go to
http://awayfind.com/avdi

#11 - 09/30/2013 08:24 PM - ko1 (Koichi Sasada)

- Category set to lib

- Status changed from Open to Feedback

- Target version set to 2.6

http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/56421
any comments?

#12 - 11/07/2014 04:19 PM - nobu (Nobuyoshi Nakada)

- Description updated

#13 - 11/07/2014 04:20 PM - nobu (Nobuyoshi Nakada)

- Related to Bug #10485: NoMethodError "undefined method `initialize_copy'" when trying to execute Queue#dup added

#14 - 01/31/2017 09:44 AM - ko1 (Koichi Sasada)

- Status changed from Feedback to Closed

No discussion.

Files

361.patch	3.34 KB	08/15/2013	zzak (zzak _)