

Ruby trunk - Feature #11242

[PATCH] socket: allow explicit buffer for recv and recv_nonblock

06/10/2015 12:43 AM - normalperson (Eric Wong)

Status:	Closed
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	

Description

Note, this relies on [Feature #11229]

This reduces GC overhead and makes the API more consistent with IO#read and IO#read_nonblock.

Benchmark results:

	user	system	total	real
alloc	0.130000	0.280000	0.410000	(0.420656)
extbuf	0.100000	0.220000	0.320000	(0.318708)

```
require 'socket'
require 'benchmark'
nr = 100000
msg = ' ' * 16384
size = msg.bytesize
buf = ' ' * size
UNIXSocket.pair(:DGRAM) do |a, b|
  Benchmark.bmbm do |x|
    x.report('alloc') do
      nr.times do
        b.send(msg, 0)
        a.recv(size, 0)
      end
    end

    x.report('extbuf') do
      nr.times do
        b.send(msg, 0)
        a.recv(size, 0, buf)
      end
    end
  end
end
```

Associated revisions

Revision a02a3f46 - 06/15/2015 08:02 PM - normal

socket: allow explicit buffer for recv and recv_nonblock

This reduces GC overhead and makes the API more consistent with IO#read and IO#read_nonblock.

- ext/socket/basicsocket.c (bsock_recv): document outbuf
- ext/socket/unixsocket.c (unix_recvfrom): ditto
- ext/socket/init.c (rsock_strbuf, recvfrom_locktmp): new functions (rsock_s_recvfrom): support destination buffer as 3rd arg (rsock_s_recvfrom_nonblock): ditto
- string.c (rb_str_locktmp_ensure): export for internal ext
- test/socket/test_nonblock.rb: test recv_nonblock
- test/socket/test_unix.rb: test recv [ruby-core:69543] [Feature #11242]

Benchmark results:

```
      user      system      total      real
```

```
alloc 0.130000 0.280000 0.410000 ( 0.420656)
extbuf 0.100000 0.220000 0.320000 ( 0.318708)
```

```
-----8<-----
require 'socket'
require 'benchmark'
nr = 100000
msg = '*' * 16384
size = msg.bytesize
buf = '*' * size
UNIXSocket.pair(:DGRAM) do |a, b|
  Benchmark.bmbm do |x|
    x.report('alloc') do
      nr.times do
        b.send(msg, 0)
        a.recv(size, 0)
      end
    end

    x.report('extbuf') do
      nr.times do
        b.send(msg, 0)
        a.recv(size, 0, buf)
      end
    end
  end
end
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@50912 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 50912 - 06/15/2015 08:02 PM - normalperson (Eric Wong)

socket: allow explicit buffer for recv and recv_nonblock

This reduces GC overhead and makes the API more consistent with IO#read and IO#read_nonblock.

- ext/socket/basicsocket.c (bsock_recv): document outbuf
- ext/socket/unixsocket.c (unix_recvfrom): ditto
- ext/socket/init.c (rsock_strbuf, recvfrom_locktmp): new functions (rsock_s_recvfrom): support destination buffer as 3rd arg (rsock_s_recvfrom_nonblock): ditto
- string.c (rb_str_locktmp_ensure): export for internal ext
- test/socket/test_nonblock.rb: test recv_nonblock
- test/socket/test_unix.rb: test recv [ruby-core:69543] [Feature #11242]

Benchmark results:

```
      user      system      total      real
```

```
alloc 0.130000 0.280000 0.410000 ( 0.420656)
extbuf 0.100000 0.220000 0.320000 ( 0.318708)
```

```
-----8<-----
require 'socket'
require 'benchmark'
nr = 100000
msg = '*' * 16384
size = msg.bytesize
buf = '*' * size
UNIXSocket.pair(:DGRAM) do |a, b|
  Benchmark.bmbm do |x|
    x.report('alloc') do
      nr.times do
        b.send(msg, 0)
        a.recv(size, 0)
      end
    end

    x.report('extbuf') do
      nr.times do
        b.send(msg, 0)
        a.recv(size, 0, buf)
      end
    end
  end
end
```

```
end
end
```

```
end
end
```

Revision 50912 - 06/15/2015 08:02 PM - normal

socket: allow explicit buffer for recv and recv_nonblock

This reduces GC overhead and makes the API more consistent with IO#read and IO#read_nonblock.

- ext/socket/basicsocket.c (bsock_recv): document outbuf
- ext/socket/unixsocket.c (unix_recvfrom): ditto
- ext/socket/init.c (rsock_strbuf, recvfrom_locktmp): new functions (rsock_s_recvfrom): support destination buffer as 3rd arg (rsock_s_recvfrom_nonblock): ditto
- string.c (rb_str_locktmp_ensure): export for internal ext
- test/socket/test_nonblock.rb: test recv_nonblock
- test/socket/test_unix.rb: test recv [ruby-core:69543] [Feature #11242]

Benchmark results:

	user	system	total	real
alloc	0.130000	0.280000	0.410000 (0.420656)	
extbuf	0.100000	0.220000	0.320000 (0.318708)	

```
-----8<-----
require 'socket'
require 'benchmark'
nr = 100000
msg = '*' * 16384
size = msg.bytesize
buf = '*' * size
UNIXSocket.pair(:DGRAM) do |a, b|
  Benchmark.bmbm do |x|
    x.report('alloc') do
      nr.times do
        b.send(msg, 0)
        a.recv(size, 0)
      end
    end

    x.report('extbuf') do
      nr.times do
        b.send(msg, 0)
        a.recv(size, 0, buf)
      end
    end
  end
end

end
end
```

Revision 50912 - 06/15/2015 08:02 PM - normal

socket: allow explicit buffer for recv and recv_nonblock

This reduces GC overhead and makes the API more consistent with IO#read and IO#read_nonblock.

- ext/socket/basicsocket.c (bsock_recv): document outbuf
- ext/socket/unixsocket.c (unix_recvfrom): ditto
- ext/socket/init.c (rsock_strbuf, recvfrom_locktmp): new functions (rsock_s_recvfrom): support destination buffer as 3rd arg (rsock_s_recvfrom_nonblock): ditto
- string.c (rb_str_locktmp_ensure): export for internal ext
- test/socket/test_nonblock.rb: test recv_nonblock
- test/socket/test_unix.rb: test recv [ruby-core:69543] [Feature #11242]

Benchmark results:

	user	system	total	real
alloc	0.130000	0.280000	0.410000 (0.420656)	
extbuf	0.100000	0.220000	0.320000 (0.318708)	

```

-----8<-----
require 'socket'
require 'benchmark'
nr = 100000
msg = '*' * 16384
size = msg.bytesize
buf = '*' * size
UNIXSocket.pair(:DGRAM) do |a, b|
Benchmark.bmbm do |x|
x.report('alloc') do
nr.times do
b.send(msg, 0)
a.recv(size, 0)
end
end

x.report('extbuf') do
nr.times do
b.send(msg, 0)
a.recv(size, 0, buf)
end
end

end
end

```

Revision 50912 - 06/15/2015 08:02 PM - normal

socket: allow explicit buffer for recv and recv_nonblock

This reduces GC overhead and makes the API more consistent with IO#read and IO#read_nonblock.

- ext/socket/basicsocket.c (bsock_recv): document outbuf
- ext/socket/unixsocket.c (unix_recvfrom): ditto
- ext/socket/init.c (rsock_strbuf, recvfrom_locktmp): new functions (rsock_s_recvfrom): support destination buffer as 3rd arg (rsock_s_recvfrom_nonblock): ditto
- string.c (rb_str_locktmp_ensure): export for internal ext
- test/socket/test_nonblock.rb: test recv_nonblock
- test/socket/test_unix.rb: test recv [ruby-core:69543] [Feature #11242]

Benchmark results:

	user	system	total	real
alloc	0.130000	0.280000	0.410000 (0.420656)	
extbuf	0.100000	0.220000	0.320000 (0.318708)	

```

-----8<-----
require 'socket'
require 'benchmark'
nr = 100000
msg = '*' * 16384
size = msg.bytesize
buf = '*' * size
UNIXSocket.pair(:DGRAM) do |a, b|
Benchmark.bmbm do |x|
x.report('alloc') do
nr.times do
b.send(msg, 0)
a.recv(size, 0)
end
end

x.report('extbuf') do
nr.times do
b.send(msg, 0)
a.recv(size, 0, buf)
end
end

end
end

```

Revision 50912 - 06/15/2015 08:02 PM - normal

socket: allow explicit buffer for recv and recv_nonblock

This reduces GC overhead and makes the API more consistent with IO#read and IO#read_nonblock.

- ext/socket/basicsocket.c (bsock_recv): document outbuf
- ext/socket/unixsocket.c (unix_recvfrom): ditto
- ext/socket/init.c (rsock_strbuf, recvfrom_locktmp): new functions (rsock_s_recvfrom): support destination buffer as 3rd arg (rsock_s_recvfrom_nonblock): ditto
- string.c (rb_str_locktmp_ensure): export for internal ext
- test/socket/test_nonblock.rb: test recv_nonblock
- test/socket/test_unix.rb: test recv [ruby-core:69543] [Feature #11242]

Benchmark results:

	user	system	total	real
alloc	0.130000	0.280000	0.410000 (0.420656)	
extbuf	0.100000	0.220000	0.320000 (0.318708)	

```

-----8<-----
require 'socket'
require 'benchmark'
nr = 100000
msg = '*' * 16384
size = msg.bytesize
buf = '*' * size
UNIXSocket.pair(:DGRAM) do |a, b|
  Benchmark.bmbm do |x|
    x.report('alloc') do
      nr.times do
        b.send(msg, 0)
        a.recv(size, 0)
      end
    end

    x.report('extbuf') do
      nr.times do
        b.send(msg, 0)
        a.recv(size, 0, buf)
      end
    end
  end
end
end
end

```

Revision d51065b0 - 06/15/2015 09:28 PM - normal

NEWS: update for socket API changes [ci skip]

[Feature #11242] [Feature #11229]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@50913 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 50913 - 06/15/2015 09:28 PM - normalperson (Eric Wong)

NEWS: update for socket API changes [ci skip]

[Feature #11242] [Feature #11229]

Revision 50913 - 06/15/2015 09:28 PM - normal

NEWS: update for socket API changes [ci skip]

[Feature #11242] [Feature #11229]

Revision 50913 - 06/15/2015 09:28 PM - normal

NEWS: update for socket API changes [ci skip]

[Feature #11242] [Feature #11229]

Revision 50913 - 06/15/2015 09:28 PM - normal

NEWS: update for socket API changes [ci skip]

[Feature #11242] [Feature #11229]

Revision 50913 - 06/15/2015 09:28 PM - normal

NEWS: update for socket API changes [ci skip]

[Feature #11242] [Feature #11229]

Revision 55a692bb - 11/17/2015 02:29 AM - normal

socket: update doc for recvfrom_nonblock [ci skip]

- ext/socket/lib/socket.rb (Socket#recvfrom_nonblock): UDPSocket#recvfrom_nonblock): update doc for exception: false and destination buffer [ruby-core:69542] [Feature #11229] [ruby-core:69543] [Feature #11242]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@52608 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 52608 - 11/17/2015 02:29 AM - normalperson (Eric Wong)

socket: update doc for recvfrom_nonblock [ci skip]

- ext/socket/lib/socket.rb (Socket#recvfrom_nonblock): UDPSocket#recvfrom_nonblock): update doc for exception: false and destination buffer [ruby-core:69542] [Feature #11229] [ruby-core:69543] [Feature #11242]

Revision 52608 - 11/17/2015 02:29 AM - normal

socket: update doc for recvfrom_nonblock [ci skip]

- ext/socket/lib/socket.rb (Socket#recvfrom_nonblock): UDPSocket#recvfrom_nonblock): update doc for exception: false and destination buffer [ruby-core:69542] [Feature #11229] [ruby-core:69543] [Feature #11242]

Revision 52608 - 11/17/2015 02:29 AM - normal

socket: update doc for recvfrom_nonblock [ci skip]

- ext/socket/lib/socket.rb (Socket#recvfrom_nonblock): UDPSocket#recvfrom_nonblock): update doc for exception: false and destination buffer [ruby-core:69542] [Feature #11229] [ruby-core:69543] [Feature #11242]

Revision 52608 - 11/17/2015 02:29 AM - normal

socket: update doc for recvfrom_nonblock [ci skip]

- ext/socket/lib/socket.rb (Socket#recvfrom_nonblock): UDPSocket#recvfrom_nonblock): update doc for exception: false and destination buffer [ruby-core:69542] [Feature #11229] [ruby-core:69543] [Feature #11242]

Revision 52608 - 11/17/2015 02:29 AM - normal

socket: update doc for recvfrom_nonblock [ci skip]

- ext/socket/lib/socket.rb (Socket#recvfrom_nonblock): UDPSocket#recvfrom_nonblock): update doc for exception: false and destination buffer [ruby-core:69542] [Feature #11229] [ruby-core:69543] [Feature #11242]

History

#1 - 06/10/2015 01:12 AM - nobu (Nobuyoshi Nakada)

- *Description updated*

#2 - 06/12/2015 08:58 AM - matz (Yukihiro Matsumoto)

Looks good to me. Go ahead.

Matz.

#3 - 06/15/2015 08:03 PM - Anonymous

- *Status changed from Open to Closed*

Applied in changeset [r50912](#).

socket: allow explicit buffer for recv and recv_nonblock

This reduces GC overhead and makes the API more consistent with IO#read and IO#read_nonblock.

- ext/socket/basicsocket.c (bsock_recv): document outbuf
- ext/socket/unixsocket.c (unix_recvfrom): ditto
- ext/socket/init.c (rsock_strbuf, recvfrom_locktmp): new functions (rsock_s_recvfrom): support destination buffer as 3rd arg (rsock_s_recvfrom_nonblock): ditto
- string.c (rb_str_locktmp_ensure): export for internal ext
- test/socket/test_nonblock.rb: test recv_nonblock
- test/socket/test_unix.rb: test recv [ruby-core:69543] [Feature [#11242](#)]

Benchmark results:

	user	system	total	real
alloc	0.130000	0.280000	0.410000 (0.420656)	
extbuf	0.100000	0.220000	0.320000 (0.318708)	

```
-----8<-----
require 'socket'
require 'benchmark'
nr = 100000
msg = '*' * 16384
size = msg.bytesize
buf = '*' * size
UNIXSocket.pair(:DGRAM) do |a, b|
  Benchmark.bmbm do |x|
    x.report('alloc') do
      nr.times do
        b.send(msg, 0)
        a.recv(size, 0)
      end
    end

    x.report('extbuf') do
      nr.times do
        b.send(msg, 0)
        a.recv(size, 0, buf)
      end
    end
  end
end
```

Files

0001-socket-allow-explicit-buffer-for-recv-and-recv_nonbl.patch	7.45 KB	06/10/2015	normalperson (Eric Wong)
---	---------	------------	--------------------------