

IO#read/gets can be very slow in doze

Status:	Closed	Backport:
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 1.9.2dev (2010-02-12 trunk 26649) [i386-mingw32]	

```
=begin
this code
```

```
n = 100 * 1000 * 1000
puts "writing"
File.open("foo", 'wb'){|f| f.write(" " * n) }
puts "reading"
File.open("foo", 'r') do |io|
  io.read
end
```

takes something like 700s in windows, most of the time is spent in `rb_str_resize`. Cause seems to be that it is `rb_str_resize`'ing the string by 1024B at a time, so N^2 in the amount of time it takes to read it in because it has to keep copying it to the newly sized string.

The same thing happens for

```
File.open("foo", 'rb') do |io|
  io.gets
end
```

except that it reads (and resizes) in 8K bytes. Not sure why they're different sizes. I'd imagine this problem exists in Linux, but without as much of a negative impact as it reads in larger sizes.

=end

Related to Ruby - Bug #1332: Reading file on Windows is 500x slower then with...	Closed	03/30/2009
--	--------	------------

#1 - 02/13/2010 12:32 PM - rogerdpack (Roger Pack)

=begin
This seems to help, though I'm not sure if it's the best way.

Index: io.c

```

--- io.c      (revision 26655)
+++ io.c      (working copy)
@@ -1170,7 +1170,7 @@ @@
if (fptr->rbuf == NULL) {
fptr->rbuf_off = 0;
fptr->rbuf_len = 0;

• [REDACTED]

• [REDACTED]
    fptr->rbuf = ALLOC_N(char, fptr->rbuf_capa);
}
if (fptr->rbuf_len == 0) {
    @@ -1583,7 +1583,7 @@ @@
    rb_exc_raise(rb_econv_open_exc(sname, dname, ecflags));

```

```
fptr->cbuf_off = 0;
fptr->cbuf_len = 0;
```

- [REDACTED]

- [REDACTED]

```
fptr->cbuf = ALLOC_N(char, fptr->cbuf_capa);
```

```
}
}
```

=end

#2 - 03/21/2010 06:07 PM - nobu (Nobuyoshi Nakada)

- *Status changed from Open to Closed*

- *% Done changed from 0 to 100*

=begin

This issue was solved with changeset r26997.

Roger, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

=end