Ruby - Bug #6401

Windows bug with File.pos

05/05/2012 12:33 AM - jmthomas (Jason Thomas)

Status:	Closed		
Priority:	Normal		
Assignee:	h.shirosaki (Hiroshi Shirosaki)		
Target version:	1.9.3		
ruby -v:	ruby 1.9.3p194 (2012-04-20) [i386-mingw32]	**Backport:**	

Description

On Windows since Ruby 1.9.3p125 there have been issues with File.pos and File.readline. Ruby 1.9.3p0 does not have this issue. I have created the following test:

```
def test_pos_with_readline
t = make_tempfile
random = Random.new(1234)
open(t.path, "w") do |f|
500.times do
f.puts "X"*random.rand(80)
end
end
i = 0
lines = open(t.path,'r').read.split("\n")
open(t.path, "r") do |f|
lines.length.times do
f.pos
assert_equal lines[i], f.readline.chomp
i += 1
end
end
end
```

If you comment out the f.pos line this test case will pass. I originally submitted issue #6179 but the fixes applied there made things better but did not complete solve the problem. I apologize for the test case but it requires many lines with newlines to reproduce.

Associated revisions

Revision 7e134cf71e2535b562cd2d81e3ff69cbe49d0ce0 - 05/08/2012 12:01 PM - h.shirosaki (Hiroshi Shirosaki)

- io.c (io_unread): fix IO#pos with mode 'r' bug on Windows.
  If the end of reading buffer is CR, io_unread() needs to unread one
  more byte.
  [ruby-core:44874] [Bug #6401]

- test/ruby/test_io_m17n.rb (TestIO_M17N#test_pos_with_buffer_end_cr):
  add a test for above.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35594 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 7e134cf7 - 05/08/2012 12:01 PM - h.shirosaki (Hiroshi Shirosaki)

- io.c (io_unread): fix IO#pos with mode 'r' bug on Windows.
  If the end of reading buffer is CR, io_unread() needs to unread one
  more byte.
  [ruby-core:44874] [Bug #6401]

- test/ruby/test_io_m17n.rb (TestIO_M17N#test_pos_with_buffer_end_cr):
  add a test for above.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35594 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 64b6f2b2a14912f46c71377cb6281765072f6563 - 05/10/2012 04:05 PM - naruse (Yui NARUSE)

merge revision(s) 35594: [Backport #6422]

```
    * io.c (io_unread): fix IO#pos with mode 'r' bug on Windows.
      If the end of reading buffer is CR, io_unread() needs to unread one
      more byte.
      [ruby-core:44874] [Bug #6401]

    * test/ruby/test_io_m17n.rb (TestIO_M17N#test_pos_with_buffer_end_cr):
      add a test for above.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@35613 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 64b6f2b2 - 05/10/2012 04:05 PM - naruse (Yui NARUSE)

merge revision(s) 35594: [Backport #6422]

```
    * io.c (io_unread): fix IO#pos with mode 'r' bug on Windows.
      If the end of reading buffer is CR, io_unread() needs to unread one
      more byte.
      [ruby-core:44874] [Bug #6401]

    * test/ruby/test_io_m17n.rb (TestIO_M17N#test_pos_with_buffer_end_cr):
      add a test for above.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@35613 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 05/05/2012 12:36 AM - luislavena (Luis Lavena)

- Category set to core

- Status changed from Open to Assigned

- Assignee set to h.shirosaki (Hiroshi Shirosaki)

#2 - 05/07/2012 09:19 PM - h.shirosaki (Hiroshi Shirosaki)

- File fix_pos_with_readline.patch added

I confirmed the issue. Thanks for your test case.
If the end of reading buffer is CR, io_unread() needs to unread one more byte for CR.
I created a patch and a simplified test case for that.

#3 - 05/08/2012 01:14 AM - jmthomas (Jason Thomas)

Is there some reason that the file reading got so messed up between 193p0 and now? Was there a refactor / rewrite of this fundamental operation?

#4 - 05/08/2012 01:34 AM - luislavena (Luis Lavena)

jmthomas (Jason Thomas) wrote:

> Is there some reason that the file reading got so messed up between 193p0 and now? Was there a refactor / rewrite of this fundamental
> operation?

Short answer: yes, there was a refactor of IO on Windows that lead a speed increase in both writing and reading big files. Seems there are corner cases that weren't covered by tests.

Long answer: both 1.9.2 and 1.9.3-p0 suffered from really slow IO reading and writing of files on Windows. This was caused primarily due newline conversion was performed always, even if was no required or the content already contained newlines.

The refactoring solved that issue and covered most of the cases exposed by tests boosting general IO operations on Windows.

However, there are cases like the one you exposed weren't covered by tests and thus, failed to get solved properly.

This refactor was introduced in 1.9.3 considering there will be another full year until Ruby 2.0 gets released. Since 1.9.2 Ruby has been getting slower and slower on Windows.

Instead of waiting to 2.0 to find and fix all those performance issues, we decided to start making a more usable Ruby today.

Hope that helps to understand the reasoning of these changes.

#5 - 05/08/2012 01:52 AM - jonforums (Jon Forums)

As just one example, due to the Windows IO refactoring led primarily by Shirosaki-san, read performance improved from ~18.5s on 1.9.2/1.9.3p0 to ~1.4s on 1.9.3p125+ with one of my micro-benchmarks

http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/42686

Yes, you read that right...18.5s to 1.4s ;)

#6 - 05/08/2012 01:02 PM - jmthomas (Jason Thomas)

Sounds like changes worth waiting for! I look forward to the next 1.9.3 patch release because my application requires this bug fix. Thanks again for your good work.

#7 - 05/08/2012 09:01 PM - Anonymous

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r35594.
Jason, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- io.c (io_unread): fix IO#pos with mode 'r' bug on Windows.
  If the end of reading buffer is CR, io_unread() needs to unread one
  more byte.
  [ruby-core:44874] [Bug #6401]

- test/ruby/test_io_m17n.rb (TestIO_M17N#test_pos_with_buffer_end_cr):
  add a test for above.

#8 - 11/02/2012 12:05 AM - jmthomas (Jason Thomas)

I can confirm this bug has been fixed in Ruby 1.9.3-p286. Thanks!

Files

fix_pos_with_readline.patch	1.28 KB	05/07/2012	h.shirosaki (Hiroshi Shirosaki)