Backport193 - Backport #6179

File::pos broken in Windows 1.9.3p125

03/21/2012 01:57 AM - jmthomas (Jason Thomas)

Status:	Closed	
Priority:	Normal	
Assignee:	h.shirosaki (Hiroshi Shirosaki)	

Description

Calling the pos method on a File in the Windows version of Ruby on 1.9.3p125 moves the file pointer. Thus it can not be called without side effect.

Associated revisions

Revision af86dba2 - 03/21/2012 08:03 AM - usa (Usaku NAKAMURA)

- test/ruby/test_io.rb (TestIO#test_pos_with_getc): added. see [Bug #6179][ruby-core:43518]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35095 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 35095 - 03/21/2012 08:03 AM - usa (Usaku NAKAMURA)

- test/ruby/test_io.rb (TestIO#test_pos_with_getc): added. see [Bug #6179][ruby-core:43518]

Revision 35095 - 03/21/2012 08:03 AM - usa (Usaku NAKAMURA)

- test/ruby/test_io.rb (TestIO#test_pos_with_getc): added. see [Bug #6179][ruby-core:43518]

Revision 35095 - 03/21/2012 08:03 AM - usa (Usaku NAKAMURA)

- test/ruby/test_io.rb (TestIO#test_pos_with_getc): added. see [Bug #6179][ruby-core:43518]

Revision 35095 - 03/21/2012 08:03 AM - usa (Usaku NAKAMURA)

- test/ruby/test_io.rb (TestIO#test_pos_with_getc): added. see [Bug #6179][ruby-core:43518]

Revision 35095 - 03/21/2012 08:03 AM - usa (Usaku NAKAMURA)

- test/ruby/test_io.rb (TestIO#test_pos_with_getc): added. see [Bug #6179][ruby-core:43518]

Revision 35095 - 03/21/2012 08:03 AM - usa (Usaku NAKAMURA)

- test/ruby/test_io.rb (TestIO#test_pos_with_getc): added. see [Bug #6179][ruby-core:43518]

Revision 7cbff3b9 - 03/22/2012 02:05 PM - shirosaki

- io.c (static int io_fflush): add the definition.
  Use it in set_binary_mode_with_seek_cur().

- io.c (set_binary_mode_with_seek_cur): refactoring to split the
  content into io_unread(). Fix the possibility of buffer overflow.

- io.c (io_unread): add new implementation for Windows. Previous one
  caused invalid cursor position using IO#pos with OS text mode. New
  one fixes the bug.

- test/ruby/test_io_m17n.rb
  (TestIO_M17N#test_pos_dont_move_cursor_position): add a test for
  above bug.
  [ruby-core:43497] [Bug #6179]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35111 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 35111 - 03/22/2012 02:05 PM - shirosaki

- io.c (static int io_fflush): add the definition.
  Use it in set_binary_mode_with_seek_cur().
- io.c (set_binary_mode_with_seek_cur): refactoring to split the content into io_unread(). Fix the possibility of buffer overflow.
- io.c (io_unread): add new implementation for Windows. Previous one caused invalid cursor position using IO#pos with OS text mode. New one fixes the bug.
- test/ruby/test_io_m17n.rb
  (TestIO_M17N#test_pos_dont_move_cursor_position): add a test for above bug.
  [ruby-core:43497] [Bug #6179]

Revision 35111 - 03/22/2012 02:05 PM - shirosaki

- io.c (static int io_fflush): add the definition.
  Use it in set_binary_mode_with_seek_cur().
- io.c (set_binary_mode_with_seek_cur): refactoring to split the content into io_unread(). Fix the possibility of buffer overflow.
- io.c (io_unread): add new implementation for Windows. Previous one caused invalid cursor position using IO#pos with OS text mode. New one fixes the bug.
- test/ruby/test_io_m17n.rb
  (TestIO_M17N#test_pos_dont_move_cursor_position): add a test for above bug.
  [ruby-core:43497] [Bug #6179]

Revision 35111 - 03/22/2012 02:05 PM - shirosaki

- io.c (static int io_fflush): add the definition.
  Use it in set_binary_mode_with_seek_cur().
- io.c (set_binary_mode_with_seek_cur): refactoring to split the content into io_unread(). Fix the possibility of buffer overflow.
- io.c (io_unread): add new implementation for Windows. Previous one caused invalid cursor position using IO#pos with OS text mode. New one fixes the bug.
- test/ruby/test_io_m17n.rb
  (TestIO_M17N#test_pos_dont_move_cursor_position): add a test for above bug.
  [ruby-core:43497] [Bug #6179]

Revision 35111 - 03/22/2012 02:05 PM - shirosaki

- io.c (static int io_fflush): add the definition.

Use it in set_binary_mode_with_seek_cur().

- io.c (set_binary_mode_with_seek_cur): refactoring to split the content into io_unread(). Fix the possibility of buffer overflow.
- io.c (io_unread): add new implementation for Windows. Previous one caused invalid cursor position using IO#pos with OS text mode. New one fixes the bug.
- test/ruby/test_io_m17n.rb (TestIO_M17N#test_pos_dont_move_cursor_position): add a test for above bug.
  [ruby-core:43497] [Bug #6179]

Revision 35111 - 03/22/2012 02:05 PM - shirosaki

- io.c (static int io_fflush): add the definition.
  Use it in set_binary_mode_with_seek_cur().
- io.c (set_binary_mode_with_seek_cur): refactoring to split the content into io_unread(). Fix the possibility of buffer overflow.
- io.c (io_unread): add new implementation for Windows. Previous one caused invalid cursor position using IO#pos with OS text mode. New one fixes the bug.
- test/ruby/test_io_m17n.rb (TestIO_M17N#test_pos_dont_move_cursor_position): add a test for above bug.
  [ruby-core:43497] [Bug #6179]

Revision 35111 - 03/22/2012 02:05 PM - shirosaki

- io.c (static int io_fflush): add the definition.
  Use it in set_binary_mode_with_seek_cur().
- io.c (set_binary_mode_with_seek_cur): refactoring to split the content into io_unread(). Fix the possibility of buffer overflow.
- io.c (io_unread): add new implementation for Windows. Previous one caused invalid cursor position using IO#pos with OS text mode. New one fixes the bug.
- test/ruby/test_io_m17n.rb (TestIO_M17N#test_pos_dont_move_cursor_position): add a test for above bug.
  [ruby-core:43497] [Bug #6179]

Revision 0e84de2b - 04/14/2012 06:02 PM - naruse (Yui NARUSE)

merge revision(s) 34785,35095,35098,35111,35152: [Backport #6294]

```
    * io.c (set_binary_mode_with_seek_cur): reorder function qualifiers.
    * test/ruby/test_io.rb (TestIO#test_pos_with_getc): added.
      see [Bug #6179][ruby-core:43518]

    * test/ruby/test_io.rb (TestIO#test_pos_with_getc): updated.
      see [ruby-core:43550]

    * io.c (static int io_fflush): add the definition.
      Use it in set_binary_mode_with_seek_cur().

    * io.c (set_binary_mode_with_seek_cur): refactoring to split the
```

```
      content into io_unread(). Fix the possibility of buffer overflow.

    * io.c (io_unread): add new implementation for Windows. Previous one
      caused invalid cursor position using IO#pos with OS text mode. New
      one fixes the bug.

    * test/ruby/test_io_m17n.rb
      (TestIO_M17N#test_pos_dont_move_cursor_position): add a test for
      above bug.
      [ruby-core:43497] [Bug #6179]

    * io.c (io_unread): fixed memory leak.  report by nagachika via IRC.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@35328 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 35328 - 04/14/2012 06:02 PM - naruse (Yui NARUSE)

merge revision(s) 34785,35095,35098,35111,35152: [Backport #6294]

```
* io.c (set_binary_mode_with_seek_cur): reorder function qualifiers.
* test/ruby/test_io.rb (TestIO#test_pos_with_getc): added.
  see [Bug #6179][ruby-core:43518]

* test/ruby/test_io.rb (TestIO#test_pos_with_getc): updated.
  see [ruby-core:43550]

* io.c (static int io_fflush): add the definition.
  Use it in set_binary_mode_with_seek_cur().

* io.c (set_binary_mode_with_seek_cur): refactoring to split the
  content into io_unread(). Fix the possibility of buffer overflow.

* io.c (io_unread): add new implementation for Windows. Previous one
  caused invalid cursor position using IO#pos with OS text mode. New
  one fixes the bug.

* test/ruby/test_io_m17n.rb
  (TestIO_M17N#test_pos_dont_move_cursor_position): add a test for
  above bug.
  [ruby-core:43497] [Bug #6179]

* io.c (io_unread): fixed memory leak.  report by nagachika via IRC.
```

History

#1 - 03/21/2012 02:43 AM - daz (Dave B)

Please provide a short code example which demonstrates
how you're checking the condition.

Have you haven't coded something like:
"if pos = 12" instead of "if pos == 12"
which would accidentally set the file position?

daz

#2 - 03/21/2012 02:56 AM - luislavena (Luis Lavena)

=begin
jmthomas (Jason Thomas) wrote:

> Calling the pos method on a File in the Windows version of Ruby on 1.9.3p125 moves the file pointer. Thus it can not be called without side effect.

Are you talking about File#pos or File#pos=?

Can you create an example like the following?

1.9.3p125 :003 > File.write "foo", "This is one line\nThis is second line\n"
=> 37
1.9.3p125 :004 > f = File.open "foo"
=> #File:foo
1.9.3p125 :005 > f.pos
=> 0
1.9.3p125 :006 > f.gets

```
=> "This is one line\n"
1.9.3p125 :007 > f.pos
=> 17
1.9.3p125 :008 > f.gets
=> "This is second line\n"
1.9.3p125 :009 > f.pos
=> 37
1.9.3p125 :010 > f.close
=> nil
1.9.3p125 :011 > exit
=end
```

#3 - 03/21/2012 02:56 AM - luislavena (Luis Lavena)

- Category set to core

- Status changed from Open to Feedback

- Priority changed from 5 to Normal

#4 - 03/21/2012 03:31 PM - phasis68 (Heesob Park)

I am not sure this issue is same to the following behavior.
But I found a bug of File#pos in trunk version.

```
C:\work>type foo.txt
1234567890

C:\work>ruby -v
ruby 2.0.0dev (2012-03-20 trunk 35094) [i386-mswin32_100]

C:\work>irb
irb(main):001:0> a = File.open('foo.txt')
=> #File:foo.txt
irb(main):002:0> a.pos
=> 0
irb(main):003:0> a.getc
=> "1"
irb(main):004:0> a.pos
=> 2
irb(main):005:0> a.getc
=> "3"
irb(main):006:0> a.pos
=> 4
irb(main):007:0> a.getc
=> "5"
irb(main):008:0> a.pos
=> 6
```

Whereas ruby 1.9.3p125 works as expected.

```
C:\work>ruby -v
ruby 1.9.3p125 (2012-02-16) [i386-mingw32]

C:\work>irb
irb(main):001:0> a = File.open('foo.txt')
=> #File:foo.txt
irb(main):002:0> a.pos
=> 0
irb(main):003:0> a.getc
=> "1"
irb(main):004:0> a.pos
=> 1
irb(main):005:0> a.getc
=> "2"
irb(main):006:0> a.pos
=> 2
irb(main):007:0> a.getc
=> "3"
irb(main):008:0> a.pos
=> 3
```

#5 - 03/21/2012 05:03 PM - usa (Usaku NAKAMURA)

- Status changed from Feedback to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r35095.
Jason, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- test/ruby/test_io.rb (TestIO#test_pos_with_getc): added. see [Bug #6179][ruby-core:43518]

#6 - 03/21/2012 05:23 PM - usa (Usaku NAKAMURA)

Hello,

In message "[ruby-core:43518] [ruby-trunk - Bug #6179] File::pos broken in Windows 1.9.3p125"
on Mar.21,2012 15:31:41, phasis@gmail.com wrote:

```
C:\work>irb
irb(main):001:0> a = File.open('foo.txt')
=> #File:foo.txt
irb(main):002:0> a.pos
=> 0
irb(main):003:0> a.getc
=> "1"
irb(main):004:0> a.pos
=> 2
irb(main):005:0> a.getc
=> "3"
irb(main):006:0> a.pos
=> 4
irb(main):007:0> a.getc
=> "5"
irb(main):008:0> a.pos
=> 6
```

Hmm, I can't reporoduce it with
ruby -v: ruby 2.0.0dev (2012-03-20 trunk 35087) [x64-mswin64_100]

I've added a test for this at r35095.

Regards,
--
U.Nakamura usa@garbagecollect.jp

#7 - 03/21/2012 08:09 PM - phasis68 (Heesob Park)

I guess it is related with binary mode reading.
If I open a file with text mode, it works fine.

```
irb(main):001:0> a = File.open('foo.txt')
=> #File:foo.txt
irb(main):002:0> a.getc
=> "1"
irb(main):003:0> a.pos
=> 2
irb(main):004:0> a.getc
=> "3"
irb(main):005:0> a.pos
fptr->rbuf.len = 0
=> 4
irb(main):006:0> a = File.open('foo.txt','rt')
=> #File:foo.txt
irb(main):007:0> a.getc
=> "1"
irb(main):008:0> a.pos
fptr->rbuf.len = 0
=> 1
irb(main):009:0> a.getc
=> "2"
irb(main):010:0> a.pos
fptr->rbuf.len = 0
=> 2
irb(main):011:0> a.getc
=> "3"
```

irb(main):012:0> a.pos
fptr->rbuf.len = 0
=> 3

BTW, my test OS is Windows XP SP3.

#8 - 03/21/2012 10:37 PM - jmthomas (Jason Thomas)

This is an issue on 1.9.3p125 on Windows and doesn't just affect binary mode. I ran Usaku's test case and it passes. However my test case fails! It appears the issue only presents itself when you try to read multiple characters? Try the following test:

def test_file_pos
File.open("test.txt", "w") {|file| file.write("Ruby's Windows\nFile::pos shouldn't\nbe broken\n")}
lines = ["Ruby's Windows\n", "File::pos shouldn't\n", "be broken\n"]
File.open("test.txt", "r") do |file|
lines.each do |line|
file.pos # Commenting out this line causes the test to pass!!!
read_line = file.readline
puts read_line
assert_equal line, read_line
end
end
end

#9 - 03/21/2012 10:43 PM - jmthomas (Jason Thomas)

I have verified this issue doesn't affect 1.9.3p0.

#10 - 03/21/2012 11:00 PM - phasis68 (Heesob Park)

The above example also affects binary mode.
If you change "r" to "rt", it works fine on 1.9.3p125.

File.open("test.txt", "w") {|file| file.write("Jason says\nThat Ruby shouldn't\nbe broken\n")}

def test_read (call_pos)
lines = ["Jason says\n", "That Ruby shouldn't\n", "be broken\n"]
File.open("test.txt", "rt") do |file|
lines.each do |line|
file.pos if call_pos # Moves file position on Ruby 1.9.3p125 on Windows
read_line = file.readline
puts read_line
raise "Error!" if line != read_line
end
end
end

Works

test_read(false)

puts

Works

test_read(true)

#11 - 03/21/2012 11:10 PM - luislavena (Luis Lavena)

=begin
jmthomas (Jason Thomas) wrote:

> This is an issue on 1.9.3p125 on Windows and doesn't just affect binary mode. I ran Usaku's test case and it passes. However my test case fails! It appears the issue only presents itself when you try to read multiple characters? Try the following test:

Are you sure your example is correct?

File#pos returns current position:

V:>irb
irb(main):001:0> File.write("foo", "line one\nline two\nline three\n")
=> 29
irb(main):002:0> f = File.open("foo")

```
=> #File:foo
irb(main):003:0> f.readline
=> "line one\n"
irb(main):004:0> f.pos
=> 10
irb(main):005:0> f.readline
=> "line two\n"
irb(main):006:0> f.pos
=> 20
irb(main):007:0> f.pos
=> 20
irb(main):008:0> f.pos
=> 20
irb(main):009:0> f.pos
=> 20
irb(main):010:0> f.readline
=> "line three\n"
irb(main):011:0> f.pos
=> 32
irb(main):012:0> f.pos
=> 32
```

Above works, f.pos is not changed *at all*.

So does using File#readlines and File#pos inside:

```
irb(main):015:0> f.readlines.each do |line|
irb(main):016:1* puts line
irb(main):017:1> f.pos
irb(main):018:1> end
line one
line two
line three
=> ["line one\n", "line two\n", "line three\n"]
=end
```

#12 - 03/21/2012 11:23 PM - phasis68 (Heesob Park)

File.open with "r" means binary mode reading and it fails.

```
C:\work>irb
irb(main):001:0> File.write("foo", "line one\nline two\nline three\n")
=> 29
irb(main):002:0> f = File.open("foo","r")
=> #File:foo
irb(main):003:0> f.readline
=> "line one\n"
irb(main):004:0> f.pos
=> 12
irb(main):005:0> f.readline
=> "ne two\n"
irb(main):006:0> f.pos
=> 21
irb(main):007:0> f.pos
=> 21
irb(main):008:0> f.readline
=> "ine three\n"
irb(main):009:0> f.pos
=> 32
```

#13 - 03/21/2012 11:30 PM - luislavena (Luis Lavena)

- Assignee set to h.shirosaki (Hiroshi Shirosaki)

phasis68 (Heesob Park) wrote:

> File.open with "r" means binary mode reading and it fails.

You mean that doing "r" (read) implies "binary"? because when open with "rb" works as expected:

```
V:>irb
irb(main):001:0> f = File.open "foo", "rb"
=> #File:foo
irb(main):002:0> f.readline
```

=> "Line one\r\n"
irb(main):003:0> f.pos
=> 10
irb(main):004:0> f.readline
=> "Line two\r\n"

But it fails when mode is implicit (only "r")

Confirmed it is a bug, assigning it.

Thank you.

#14 - 03/21/2012 11:39 PM - ryanmelt (Ryan Melton)

This bug is still marked closed. I believe it should be open now that Luis has confirmed and assigned it, correct?

#15 - 03/21/2012 11:44 PM - luislavena (Luis Lavena)

- Status changed from Closed to Assigned

Sorry, Redmine closed when Usa Nakamura committed a test for it. Reopening.

#16 - 03/21/2012 11:44 PM - luislavena (Luis Lavena)

- ruby -v changed from 1.9.3p125 to ruby 1.9.3p125 (2012-02-16) [i386-mingw32]

#17 - 03/22/2012 03:23 AM - usa (Usaku NAKAMURA)

Hello,

In message "[ruby-core:43539] [ruby-trunk - Bug #6179][Assigned] File::pos broken in Windows 1.9.3p125"
on Mar.21,2012 23:44:22, luislavena@gmail.com wrote:

> Status changed from Closed to Assigned
>
> Sorry, Redmine closed when Usa Nakamura committed a test for it. Reopening.

Oops. Sorry, and thank you luis.
Commit-Redmine gateway is too difficult...

Regards,
--
U.Nakamura usa@garbagecollect.jp

#18 - 03/22/2012 04:49 AM - h.shirosaki (Hiroshi Shirosaki)

- File pos_fix.patch added

Thank you for your work.
I confirmed the bug. io_unread() with mode "r" had a bug. IO#pos calls io_unread().

Trunk also has same bug.
I've created a patch which replaces io_unread(). Heesob's test case was passed.
make test && make test-all look fine.

The patch is against trunk.
Could you review and try the patch?

#19 - 03/22/2012 09:32 AM - phasis68 (Heesob Park)

This issue is related with new line conversion.
I think Revision 35095 made a wrong test case.

test_pos_with_getc methods should be modified like this:

```
def test_pos_with_getc
bug6179 = '[ruby-core:43497]'
t = make_tempfile
open(t.path, "w") do |f|
f.write "0123456789\n"
end

open(t.path, "r") do |f|
```

```
  assert_equal 0, f.pos
  assert_equal '0', f.getc
  assert_equal 1, f.pos
  assert_equal '1', f.getc
  assert_equal 2, f.pos
  assert_equal '2', f.getc
  assert_equal 3, f.pos
  assert_equal '3', f.getc
  assert_equal 4, f.pos
  assert_equal '4', f.getc
end
```

end

#20 - 03/22/2012 12:23 PM - usa (Usaku NAKAMURA)

Hello,

In message "[ruby-core:43542] [ruby-trunk - Bug #6179] File::pos broken in Windows 1.9.3p125"
on Mar.22,2012 04:49:22, h.shirosaki@gmail.com wrote:

> The patch is against trunk.
> Could you review and try the patch?

It seems good.

Regards,
--
U.Nakamura usa@garbagecollect.jp

#21 - 03/22/2012 11:05 PM - Anonymous

- Status changed from Assigned to Closed

This issue was solved with changeset r35111.
Jason, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- io.c (static int io_fflush): add the definition.
  Use it in set_binary_mode_with_seek_cur().
- io.c (set_binary_mode_with_seek_cur): refactoring to split the
  content into io_unread(). Fix the possibility of buffer overflow.
- io.c (io_unread): add new implementation for Windows. Previous one
  caused invalid cursor position using IO#pos with OS text mode. New
  one fixes the bug.
- test/ruby/test_io_m17n.rb
  (TestIO_M17N#test_pos_dont_move_cursor_position): add a test for
  above bug.
  [ruby-core:43497] [Bug #6179]

#22 - 03/23/2012 09:47 AM - h.shirosaki (Hiroshi Shirosaki)

I fixed it at r35111.

Please move this 1.9.3 backport.

Revisions:
34785,35095,35098,35111

#23 - 03/23/2012 10:07 AM - usa (Usaku NAKAMURA)

- Tracker changed from Bug to Backport

- Project changed from Ruby trunk to Backport193

- Category deleted (core)

- Target version deleted (1.9.3)

moved to backport93.

BTW, you can move tickets by yourself, shirosaki-san.

#24 - 04/03/2012 05:11 PM - h.shirosaki (Hiroshi Shirosaki)

- File pos_backport.patch added

usa (Usaku NAKAMURA) wrote:

> moved to backport93.
>
> BTW, you can move tickets by yourself, shirosaki-san.

Thank you.
I couldn't move this because I'm ruby-trunk member but not yet Backport93 member on Redmine.

This ticket remains closed. Should this be opened? I can't re-open this.

Add r35152 (memory leak fix. Thanks to usa-san, nagachika-san)

Revisions:
34785,35095,35098,35111,35152

I confirmed make test, make test-all was good at the following revision with the backport patch.

ruby 1.9.3p173 (2012-04-01 revision 35203) [i386-mingw32]

#25 - 05/04/2012 11:56 PM - jmthomas (Jason Thomas)

This does not appear to fixed in Ruby 1.9.3p194, however your test case passes. I have created the following failing test case:

```
def test_pos_with_readline
t = make_tempfile
random = Random.new(1234)
open(t.path, "w") do |f|
1000.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
puts f.pos
assert_equal lines[i], f.readline.chomp
i += 1
end
end
end
```

#26 - 05/05/2012 12:14 AM - luislavena (Luis Lavena)

Please open a new issue so we can discuss further. The mixing between pos and readline is not clear time and your test isn't helping in that area.

Thank you.

Files

pos_fix.patch	4.48 KB	03/22/2012	h.shirosaki (Hiroshi Shirosaki)
pos_backport.patch	5.73 KB	04/03/2012	h.shirosaki (Hiroshi Shirosaki)