Ruby master - Bug #13965

Surprising behavior when using Tempfile.open

10/03/2017 12:03 AM - davemyron (Dave Myron)

Status:	Rejected		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	2.4.1p111	**Backport:**	2.3: UNKNOWN, 2.4: UNKNOWN

Description

I was trying to write the contents of an array to a Tempfile using the #open() + block syntax: Tempfile.open { |f| my_array.each { |item| f.puts item }}

However, I was finding that only *most* of the items were being written to the file which, in my case, was Very Bad.

Witness (with Ruby 2.4.1p111; same behavior in 2.4.3):

```
>> Tempfile.open() { |f| (1..10000).each { |i| f.puts i }; `tail -n1 #{f.path}` }
=> "8416"
>> Tempfile.open() { |f| (1..100000).each { |i| f.puts i }; `tail -n1 #{f.path}` }
=> "98835"
>> Tempfile.open() { |f| f.puts "Test\n"; `tail -n1 #{f.path}` }
=> ""
```

All very surprising results! When using block syntax, I expect to be able to interact with the Tempfile inside the block and then forget about it.

Storing the path of the Tempfile and accessing it after the block is closed produces the expected results:

```
>> path = ""; Tempfile.open() { |f| path = f.path; f.puts "Test\n" }; `tail -n1 #{path}`
=> "Test\n"
>> path = ""; Tempfile.open() { |f| path = f.path; (1..10000).each { |i| f.puts i }}; `tail -n1 #{
path}`
=> "10000\n"
```

I suspect that it's due to some buffering. Nonetheless, it's unexpected to have to store the path to the Tempfile when using the block syntax and to interact with after closing the block.

Associated revisions

Revision 09e60b5a - 10/05/2017 02:35 AM - nobu (Nobuyoshi Nakada)

io.c: [DOC] about buffering [ci skip]

- io.c (rb_file_initialize): [DOC] stated that non-tty file is buffered by the default, and added links to related methods. [ruby-core:83081] [Bug #13965]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60121 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 60121 - 10/05/2017 02:35 AM - nobu (Nobuyoshi Nakada)

io.c: [DOC] about buffering [ci skip]

- io.c (rb_file_initialize): [DOC] stated that non-tty file is buffered by the default, and added links to related methods. [ruby-core:83081] [Bug #13965]

Revision 60121 - 10/05/2017 02:35 AM - nobu (Nobuyoshi Nakada)

io.c: [DOC] about buffering [ci skip]

- io.c (rb_file_initialize): [DOC] stated that non-tty file is buffered by the default, and added links to related methods. [ruby-core:83081] [Bug #13965]

Revision 60121 - 10/05/2017 02:35 AM - nobu (Nobuyoshi Nakada)

io.c: [DOC] about buffering [ci skip]

- io.c (rb_file_initialize): [DOC] stated that non-tty file is buffered by the default, and added links to related methods. [ruby-core:83081] [Bug #13965]

History

#1 - 10/03/2017 12:12 AM - davemyron (Dave Myron)

This appears to be the behavior of File.open + block, as well:

```
>> File.open("/tmp/test.txt","w+"){|f| (1..10000).each { |i| f.puts i }; `tail -n1 #{f.path}` }
=> "8416"
```

#2 - 10/03/2017 12:17 AM - davemyron (Dave Myron)

To workaround the unexpected behavior, either use IO#sync = true or IO#flush after writing to the file.

#3 - 10/03/2017 03:44 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Rejected

Yes, it's a nature of buffering.

#4 - 10/04/2017 05:52 PM - davemyron (Dave Myron)

Could a small documentation blurb alleviate future headaches for other users?

If the docs had said something like "Due to buffering, the contents of a file may be incomplete until flushed" I would not have been (so) surprised.