# Ruby master - Bug #16997

## IO#gets converts some \r\n to \n with universal_newline: false

06/27/2020 09:26 PM - scivola20 (sciv ola)

| | | | |
|---|---|---|---|
| **Status:** | Open | | |
| **Priority:** | Normal | | |
| **Assignee:** | | | |
| **Target version:** | | | |
| **ruby -v:** | ruby 2.7.1p83 (2020-03-31 revision a0c7c23c9c) [x86_64-darwin17] | **Backport:** | 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN |

**Description**

Reproduction code:

```
IO.binwrite "t.csv", ("a" * 100 + "\r\n") * 100
File.open("t.csv", encoding: "BOM|UTF-8", universal_newline: false) do |input|
  p input.gets(nil, 32 * 1024) # => "a...a\n...\na...a\r\n...\r\n"
end
```

It causes MalformedCSVError at opening CSV file with `encoding: "BOM|UTF-8":
https://github.com/ruby/csv/issues/147

### History

**#1 - 08/26/2020 05:20 PM - jeremyevans0 (Jeremy Evans)**

I'm able to reproduce this issue on Windows (ruby 2.7.0p0 (2019-12-25 revision 647ee6f091) [x64-mingw32]), but not on OpenBSD (probably expected).

On Windows, this doesn't just affect IO#gets, it also affects IO#read and likely other IO methods for reading. From some testing, it appears that the first 8KB read have the \r\n -> \n newline translation performed, and it is specific to BOM|UTF-8, it doesn't happen with just UTF-8. 8KB happens to be IO_RBUF_CAPA_MIN. My guess is the initial 8KB gets buffered before the universal newline setting is applied runs due to the BOM detection. Assuming that is the issue, there may be a couple possible solutions:

- Apply the universal newline setting before the BOM detection (seems best).
- Clear the buffer after the BOM detection and set the current file position to directly after the BOM. The next read would then fill the buffer and hopefully work correctly.

Unfortunately, I don't have a Windows development environment for Ruby, so I can't currently do more than speculate.