Ruby master - Bug #11391
chopped_length should return early for empty strings
07/23/2015 08:18 AM - cesario (Franck Verrot)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>ruby -v: ruby 2.3.0dev (2015-07-22 trunk 51338) [x86_64-darwin14]</td>
</tr>
<tr>
<td>Backport:</td>
<td></td>
</tr>
</tbody>
</table>

Description

The current implementation only tries to check if the beginning of the string isn't coming after its end (I'm not even really sure if that's possibly actually, is there case where RSTRING_LEN returns a negative number?).

In any case, here's the change (and the patch's attached to the ticket):

diff --git a/string.c b/string.c
index 5696057..592fb2c 100644
--- a/string.c
+++ b/string.c
@@ -7138,7 +7138,7 @@ chopped_length(VALUE str)
     beg = RSTRING_PTR(str);
     end = beg + RSTRING_LEN(str);
-    if (beg > end) return 0;
+    if (beg >= end) return 0;
     p = rb_enc_prev_char(beg, end, end, enc);

I ran the test and things seem to look OK :-) I hope I'm not overlooking something.

Thanks!

Associated revisions

Revision fb84b86b - 02/07/2019 07:39 AM - nobu (Nobuyoshi Nakada)

- string.c (chopped_length): early return for empty strings

[Bug #11391]

From: Franck Verrot franck@verrot.fr

Revision 67018 - 02/07/2019 07:39 AM - nobu (Nobuyoshi Nakada)

- string.c (chopped_length): early return for empty strings

[Bug #11391]

From: Franck Verrot franck@verrot.fr

History

#1 - 02/07/2019 07:39 AM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed

Applied in changeset trunk@67018.

- string.c (chopped_length): early return for empty strings

[Bug #11391]
From: Franck Verrot franck@verrot.fr

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001-string.c-chopped_length-early-return-for-empty-strin.patch</td>
<td>720 Bytes</td>
<td>07/23/2015</td>
<td>cesario (Franck Verrot)</td>
</tr>
</tbody>
</table>