

Ruby - Bug #19792

arm alpine segmentation fault

08/01/2023 04:52 AM - koke2y

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 3.2.2 (2023-03-30 revision e51014f9c0) +YJIT [aarch64-linux-musl]	Backport: 3.0: UNKNOWN, 3.1: UNKNOWN, 3.2: DONE

Description

Docker version: Docker version 24.0.2, build cb74dfc
Docker image: ruby:3.2.2-alpine3.18

- chip: Apple M2 Max
- OS version: Version 13.4

segmentation fault

```
str = 'a' * 100_000_000

hash = {
  x: {
    y: {
      z: str
    }
  }
}

p hash

81a3dc0d38e8:/api# ruby test.rb
test.rb:11: [BUG] Segmentation fault at 0x0000ffff84e454a0
ruby 3.2.2 (2023-03-30 revision e51014f9c0) +YJIT [aarch64-linux-musl]

-- Control frame information -----
c:0006 p:---- s:0022 e:000021 CFUNC :inspect
c:0005 p:---- s:0019 e:000018 CFUNC :inspect
c:0004 p:---- s:0016 e:000015 CFUNC :inspect
c:0003 p:---- s:0013 e:000012 CFUNC :p
c:0002 p:0029 s:0008 E:001cc0 EVAL test.rb:11 [FINISH]
c:0001 p:0000 s:0003 E:000da0 DUMMY [FINISH]
```

Associated revisions

Revision 2214bcb70d9f9120f1f3790ca340236c8f080991 - 08/23/2023 10:07 PM - alanwu (Alan Wu)

Fix premature string collection during append

Previously, the following crashed due to use-after-free with AArch64 Alpine Linux 3.18.3 (aarch64-linux-musl):

```
str = 'a' * (32*1024*1024)
p({z: str})
```

32 MiB is the default for GC_MALLOC_LIMIT_MAX, and the crash could be dodged by setting RUBY_GC_MALLOC_LIMIT_MAX to large values.

Under a debugger, one can see the str2 of rb_str_buf_append() getting prematurely collected while str_buf_cat4() allocates capacity.

Add GC guards so the buffer of str2 lives across the GC run initiated in str_buf_cat4().

[Bug #19792]

Revision 2214bcb70d9f9120f1f3790ca340236c8f080991 - 08/23/2023 10:07 PM - alanwu (Alan Wu)

Fix premature string collection during append

Previously, the following crashed due to use-after-free with AArch64 Alpine Linux 3.18.3 (aarch64-linux-musl):

```
str = 'a' * (32*1024*1024)
p({z: str})
```

32 MiB is the default for GC_MALLOC_LIMIT_MAX, and the crash could be dodged by setting RUBY_GC_MALLOC_LIMIT_MAX to large values. Under a debugger, one can see the str2 of rb_str_buf_append() getting prematurely collected while str_buf_cat4() allocates capacity.

Add GC guards so the buffer of str2 lives across the GC run initiated in str_buf_cat4().

[Bug #19792]

Revision 2214bcb7 - 08/23/2023 10:07 PM - alanwu (Alan Wu)

Fix premature string collection during append

Previously, the following crashed due to use-after-free with AArch64 Alpine Linux 3.18.3 (aarch64-linux-musl):

```
str = 'a' * (32*1024*1024)
p({z: str})
```

32 MiB is the default for GC_MALLOC_LIMIT_MAX, and the crash could be dodged by setting RUBY_GC_MALLOC_LIMIT_MAX to large values. Under a debugger, one can see the str2 of rb_str_buf_append() getting prematurely collected while str_buf_cat4() allocates capacity.

Add GC guards so the buffer of str2 lives across the GC run initiated in str_buf_cat4().

[Bug #19792]

Revision d30781db4de82a891712f359d7659c9fc98cb215 - 09/30/2023 04:07 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 2214bcb70d9f9120f1f3790ca340236c8f080991: [Backport #19792]

Fix premature string collection during append

Previously, the following crashed due to use-after-free with AArch64 Alpine Linux 3.18.3 (aarch64-linux-musl):

```
```ruby
str = 'a' * (32*1024*1024)
p({z: str})
```
```

32 MiB is the default for `GC\_MALLOC\_LIMIT\_MAX`, and the crash could be dodged by setting `RUBY\_GC\_MALLOC\_LIMIT\_MAX` to large values. Under a debugger, one can see the `str2` of rb_str_buf_append() getting prematurely collected while str_buf_cat4() allocates capacity.

Add GC guards so the buffer of `str2` lives across the GC run initiated in str_buf_cat4().

```
[Bug #19792]
---
string.c | 2 ++
1 file changed, 2 insertions(+)
```

merge revision(s) 2214bcb70d9f9120f1f3790ca340236c8f080991: [Backport #19792]

```
Fix premature string collection during append

Previously, the following crashed due to use-after-free
with AArch64 Alpine Linux 3.18.3 (aarch64-linux-musl):

```ruby
str = 'a' * (32*1024*1024)
p({z: str})
```

32 MiB is the default for `GC_MALLOC_LIMIT_MAX`, and the crash
could be dodged by setting `RUBY_GC_MALLOC_LIMIT_MAX` to large values.
Under a debugger, one can see the `str2` of rb_str_buf_append()
getting prematurely collected while str_buf_cat4() allocates capacity.

Add GC guards so the buffer of `str2` lives across the GC run
initiated in str_buf_cat4().

[Bug #19792]
---
string.c | 2 ++
1 file changed, 2 insertions(+)
```

merge revision(s) 2214bcb70d9f9120f1f3790ca340236c8f080991: [Backport #19792]

```
Fix premature string collection during append

Previously, the following crashed due to use-after-free
with AArch64 Alpine Linux 3.18.3 (aarch64-linux-musl):

```ruby
str = 'a' * (32*1024*1024)
p({z: str})
```

32 MiB is the default for `GC_MALLOC_LIMIT_MAX`, and the crash
could be dodged by setting `RUBY_GC_MALLOC_LIMIT_MAX` to large values.
Under a debugger, one can see the `str2` of rb_str_buf_append()
getting prematurely collected while str_buf_cat4() allocates capacity.

Add GC guards so the buffer of `str2` lives across the GC run
initiated in str_buf_cat4().

[Bug #19792]
---
string.c | 2 ++
1 file changed, 2 insertions(+)
```

History

#1 - 08/01/2023 04:58 AM - k0kubun (Takashi Kokubun)

00000000000000000000 [BUG] 000000000000000000000000000000000000? c:0001 p:0000 s:0003 E:000da0 DUMMY [FINISH]
000

#2 - 08/01/2023 05:24 AM - koke2y (00 00)

k0kubun (Takashi Kokubun) wrote in #note-1:

00000000000000000000 [BUG] 000000000000000000000000000000000000? c:0001 p:0000 s:0003 E:000da0 DUMMY [FINISH]
000

00000000000000000000
0000000000000000

test.rb:11: [BUG] Segmentation fault at 0x0000ffff84e454a0
ruby 3.2.2 (2023-03-30 revision e51014f9c0) +YJIT [aarch64-linux-musl]

-- Control frame information -----

```
c:0006 p:---- s:0022 e:000021 CFUNC :inspect
c:0005 p:---- s:0019 e:000018 CFUNC :inspect
c:0004 p:---- s:0016 e:000015 CFUNC :inspect
c:0003 p:---- s:0013 e:000012 CFUNC :p
c:0002 p:0029 s:0008 E:001cc0 EVAL test.rb:11 [FINISH]
c:0001 p:0000 s:0003 E:000da0 DUMMY [FINISH]
```

```
-- Ruby level backtrace information -----
test.rb:11:in `<main>'
test.rb:11:in `p'
test.rb:11:in `inspect'
test.rb:11:in `inspect'
test.rb:11:in `inspect'
```

```
-- Machine register context -----
x0: 0x0000ffff7ce444b5 x1: 0x0000ffff84e454a0 x2: 0x00000000005f5e102
x3: 0x0000000000000000 x4: 0x0000ffff8ada35a2 x5: 0x0000ffff82da25b7
x6: 0x000000003d7a3a7b x7: 0x0000ffff9806d410 x18: 0x000000000000007fc
x19: 0x0000000007ffffff x20: 0x0000ffff7ce444b0 x21: 0x0000000000000001
x22: 0x0000ffff9301fb60 x23: 0x00000000005f5e107 x24: 0xffffffffffffffff
x25: 0x0000ffff84e454a0 x26: 0x00000000005f5e102 x27: 0x0000000000000005
x28: 0x0000000000000005 x29: 0x0000ffffefa08b80 sp: 0x0000ffffefa08b80
fau: 0x0000ffff84e454a0
```

```
-- Other runtime information -----
```

```
* Loaded script: test.rb
```

```
* Loaded features:
```

```
0 enumerator.so
1 thread.rb
2 fiber.so
3 rational.so
4 complex.so
5 ruby2_keywords.rb
6 /usr/local/lib/ruby/3.2.0/aarch64-linux-musl/enc/encdb.so
7 /usr/local/lib/ruby/3.2.0/aarch64-linux-musl/enc/trans/transdb.so
8 /usr/local/lib/ruby/3.2.0/aarch64-linux-musl/rbconfig.rb
9 /usr/local/lib/ruby/3.2.0/rubygems/compatibility.rb
10 /usr/local/lib/ruby/3.2.0/rubygems/defaults.rb
11 /usr/local/lib/ruby/3.2.0/rubygems/deprecate.rb
12 /usr/local/lib/ruby/3.2.0/rubygems/errors.rb
13 /usr/local/lib/ruby/3.2.0/rubygems/unknown_command_spell_checker.rb
14 /usr/local/lib/ruby/3.2.0/rubygems/exceptions.rb
15 /usr/local/lib/ruby/3.2.0/rubygems/basic_specification.rb
16 /usr/local/lib/ruby/3.2.0/rubygems/stub_specification.rb
17 /usr/local/lib/ruby/3.2.0/rubygems/platform.rb
18 /usr/local/lib/ruby/3.2.0/rubygems/util/list.rb
19 /usr/local/lib/ruby/3.2.0/rubygems/version.rb
20 /usr/local/lib/ruby/3.2.0/rubygems/requirement.rb
21 /usr/local/lib/ruby/3.2.0/rubygems/specification.rb
22 /usr/local/lib/ruby/3.2.0/rubygems/util.rb
23 /usr/local/lib/ruby/3.2.0/rubygems/dependency.rb
24 /usr/local/lib/ruby/3.2.0/rubygems/core_ext/kernel_gem.rb
25 /usr/local/lib/ruby/3.2.0/aarch64-linux-musl/monitor.so
26 /usr/local/lib/ruby/3.2.0/monitor.rb
27 /usr/local/lib/ruby/3.2.0/rubygems.rb
28 /usr/local/lib/ruby/3.2.0/rubygems/path_support.rb
29 /usr/local/lib/ruby/3.2.0/error_highlight/version.rb
30 /usr/local/lib/ruby/3.2.0/error_highlight/base.rb
31 /usr/local/lib/ruby/3.2.0/error_highlight/formatter.rb
32 /usr/local/lib/ruby/3.2.0/error_highlight/core_ext.rb
33 /usr/local/lib/ruby/3.2.0/error_highlight.rb
34 /usr/local/lib/ruby/3.2.0/did_you_mean/version.rb
35 /usr/local/lib/ruby/3.2.0/did_you_mean/core_ext/name_error.rb
36 /usr/local/lib/ruby/3.2.0/did_you_mean/levenshtein.rb
37 /usr/local/lib/ruby/3.2.0/did_you_mean/jaro_winkler.rb
38 /usr/local/lib/ruby/3.2.0/did_you_mean/spell_checker.rb
39 /usr/local/lib/ruby/3.2.0/did_you_mean/spell_checkers/name_error_checkers/class_name_checker.rb
40 /usr/local/lib/ruby/3.2.0/did_you_mean/spell_checkers/name_error_checkers/variable_name_checker.rb
41 /usr/local/lib/ruby/3.2.0/did_you_mean/spell_checkers/name_error_checkers.rb
42 /usr/local/lib/ruby/3.2.0/did_you_mean/spell_checkers/method_name_checker.rb
43 /usr/local/lib/ruby/3.2.0/did_you_mean/spell_checkers/key_error_checker.rb
44 /usr/local/lib/ruby/3.2.0/did_you_mean/spell_checkers/null_checker.rb
```

```

45 /usr/local/lib/ruby/3.2.0/did_you_mean/tree_spell_checker.rb
46 /usr/local/lib/ruby/3.2.0/did_you_mean/spell_checkers/require_path_checker.rb
47 /usr/local/lib/ruby/3.2.0/did_you_mean/spell_checkers/pattern_key_name_checker.rb
48 /usr/local/lib/ruby/3.2.0/did_you_mean/formatter.rb
49 /usr/local/lib/ruby/3.2.0/did_you_mean.rb
50 /usr/local/lib/ruby/3.2.0/syntax_suggest/core_ext.rb

```

* Process memory map:

```

aaaac3bd0000-aaaac3bd1000 r-xp 00000000 fe:01 2507891 /usr/local/bin/ruby
aaaac3bef000-aaaac3bf0000 r--p 0000f000 fe:01 2507891 /usr/local/bin/ruby
aaaac3bf0000-aaaac3bf1000 rw-p 00010000 fe:01 2507891 /usr/local/bin/ruby
aaaaf32b4000-aaaaf32b5000 ---p 00000000 00:00 0 [heap]
aaaaf32b5000-aaaaf32c1000 rw-p 00000000 00:00 0 [heap]
ffff7ce44000-ffff84e45000 rw-p 00000000 00:00 0
ffff8ce46000-ffff92da9000 rw-p 00000000 00:00 0
ffff92dab000-ffff92db7000 rw-p 00000000 00:00 0
ffff92db7000-ffff92dd6000 r-xp 00000000 fe:01 2508203 /usr/local/lib/ruby/3.2.0/aarch64-lin
ux-musl/monitor.so
ffff92dd6000-ffff92dd7000 r--p 0000f000 fe:01 2508203 /usr/local/lib/ruby/3.2.0/aarch64-lin
ux-musl/monitor.so
ffff92dd7000-ffff92dd8000 rw-p 00010000 fe:01 2508203 /usr/local/lib/ruby/3.2.0/aarch64-lin
ux-musl/monitor.so
ffff92dd8000-ffff92e40000 rw-p 00000000 00:00 0
ffff92e44000-ffff92e80000 rw-p 00000000 00:00 0
ffff92e82000-ffff92ebe000 rw-p 00000000 00:00 0
ffff92ec4000-ffff92ecc000 rw-p 00000000 00:00 0
ffff92ed0000-ffff92f64000 rw-p 00000000 00:00 0
ffff92f66000-ffff93185000 rw-p 00000000 00:00 0
ffff93185000-ffff931a4000 r-xp 00000000 fe:01 2508176 /usr/local/lib/ruby/3.2.0/aarch64-lin
ux-musl/enc/trans/transdb.so
ffff931a4000-ffff931a5000 r--p 0000f000 fe:01 2508176 /usr/local/lib/ruby/3.2.0/aarch64-lin
ux-musl/enc/trans/transdb.so
ffff931a5000-ffff931a6000 rw-p 00010000 fe:01 2508176 /usr/local/lib/ruby/3.2.0/aarch64-lin
ux-musl/enc/trans/transdb.so
ffff931a6000-ffff931aa000 rw-p 00000000 00:00 0
ffff931aa000-ffff931c9000 r-xp 00000000 fe:01 2508132 /usr/local/lib/ruby/3.2.0/aarch64-lin
ux-musl/enc/encdb.so
ffff931c9000-ffff931ca000 r--p 0000f000 fe:01 2508132 /usr/local/lib/ruby/3.2.0/aarch64-lin
ux-musl/enc/encdb.so
ffff931ca000-ffff931cb000 rw-p 00010000 fe:01 2508132 /usr/local/lib/ruby/3.2.0/aarch64-lin
ux-musl/enc/encdb.so
ffff931cb000-ffff931d3000 rw-p 00000000 00:00 0
ffff931d3000-ffff931d4000 ---p 00000000 00:00 0
ffff931d4000-ffff93275000 rw-p 00000000 00:00 0
ffff93275000-ffff93276000 ---p 00000000 00:00 0
ffff93276000-ffff93317000 rw-p 00000000 00:00 0
ffff93317000-ffff93318000 ---p 00000000 00:00 0
ffff93318000-ffff933b9000 rw-p 00000000 00:00 0
ffff933b9000-ffff933ba000 ---p 00000000 00:00 0
ffff933ba000-ffff9345b000 rw-p 00000000 00:00 0
ffff9345b000-ffff9345c000 ---p 00000000 00:00 0
ffff9345c000-ffff934fd000 rw-p 00000000 00:00 0
ffff934fd000-ffff934fe000 ---p 00000000 00:00 0
ffff934fe000-ffff9359f000 rw-p 00000000 00:00 0
ffff9359f000-ffff935a0000 ---p 00000000 00:00 0
ffff935a0000-ffff93641000 rw-p 00000000 00:00 0
ffff93641000-ffff93642000 ---p 00000000 00:00 0
ffff93642000-ffff936e3000 rw-p 00000000 00:00 0
ffff936e3000-ffff936e4000 ---p 00000000 00:00 0
ffff936e4000-ffff93785000 rw-p 00000000 00:00 0
ffff93785000-ffff93786000 ---p 00000000 00:00 0
ffff93786000-ffff93827000 rw-p 00000000 00:00 0
ffff93827000-ffff93828000 ---p 00000000 00:00 0
ffff93828000-ffff938c9000 rw-p 00000000 00:00 0
ffff938c9000-ffff938ca000 ---p 00000000 00:00 0
ffff938ca000-ffff9396b000 rw-p 00000000 00:00 0
ffff9396b000-ffff9396c000 ---p 00000000 00:00 0
ffff9396c000-ffff93a0d000 rw-p 00000000 00:00 0
ffff93a0d000-ffff93a0e000 ---p 00000000 00:00 0
ffff93a0e000-ffff93aaf000 rw-p 00000000 00:00 0
ffff93aaf000-ffff93ab0000 ---p 00000000 00:00 0
ffff93ab0000-ffff93b51000 rw-p 00000000 00:00 0
ffff93b51000-ffff93b52000 ---p 00000000 00:00 0
ffff93b52000-ffff93bf3000 rw-p 00000000 00:00 0

```

```

fffff93bf3000-fffff93bf4000 ---p 00000000 00:00 0
fffff93bf4000-fffff93c95000 rw-p 00000000 00:00 0
fffff93c95000-fffff93c96000 ---p 00000000 00:00 0
fffff93c96000-fffff93d37000 rw-p 00000000 00:00 0
fffff93d37000-fffff93d38000 ---p 00000000 00:00 0
fffff93d38000-fffff93dd9000 rw-p 00000000 00:00 0
fffff93dd9000-fffff93dda000 ---p 00000000 00:00 0
fffff93dda000-fffff93e7b000 rw-p 00000000 00:00 0
fffff93e7b000-fffff93e7c000 ---p 00000000 00:00 0
fffff93e7c000-fffff93f1d000 rw-p 00000000 00:00 0
fffff93f1d000-fffff93f1e000 ---p 00000000 00:00 0
fffff93f1e000-fffff93fbf000 rw-p 00000000 00:00 0
fffff93fbf000-fffff93fc0000 ---p 00000000 00:00 0
fffff93fc0000-fffff94061000 rw-p 00000000 00:00 0
fffff94061000-fffff94062000 ---p 00000000 00:00 0
fffff94062000-fffff94103000 rw-p 00000000 00:00 0
fffff94103000-fffff94104000 ---p 00000000 00:00 0
fffff94104000-fffff941a5000 rw-p 00000000 00:00 0
fffff941a5000-fffff941a6000 ---p 00000000 00:00 0
fffff941a6000-fffff94247000 rw-p 00000000 00:00 0
fffff94247000-fffff94248000 ---p 00000000 00:00 0
fffff94248000-fffff942e9000 rw-p 00000000 00:00 0
fffff942e9000-fffff942ea000 ---p 00000000 00:00 0
fffff942ea000-fffff9438b000 rw-p 00000000 00:00 0
fffff9438b000-fffff9438c000 ---p 00000000 00:00 0
fffff9438c000-fffff9442d000 rw-p 00000000 00:00 0
fffff9442d000-fffff9442e000 ---p 00000000 00:00 0
fffff9442e000-fffff944cf000 rw-p 00000000 00:00 0
fffff944cf000-fffff944d0000 ---p 00000000 00:00 0
fffff944d0000-fffff94571000 rw-p 00000000 00:00 0
fffff94571000-fffff94572000 ---p 00000000 00:00 0
fffff94572000-fffff978c6000 rw-p 00000000 00:00 0
fffff978c6000-fffff978f5000 r-xp 00000000 fe:01 2507563 /usr/lib/libgcc_s.so.1
fffff978f5000-fffff978f6000 r--p 0001f000 fe:01 2507563 /usr/lib/libgcc_s.so.1
fffff978f6000-fffff978f7000 rw-p 00020000 fe:01 2507563 /usr/lib/libgcc_s.so.1
fffff978f7000-fffff97975000 r-xp 00000000 fe:01 2507567 /usr/lib/libgmp.so.10.4.1
fffff97975000-fffff97977000 r--p 0006e000 fe:01 2507567 /usr/lib/libgmp.so.10.4.1
fffff97977000-fffff97978000 rw-p 00070000 fe:01 2507567 /usr/lib/libgmp.so.10.4.1
fffff97978000-fffff979a7000 r-xp 00000000 fe:01 2504591 /lib/libz.so.1.2.13
fffff979a7000-fffff979a8000 r--p 0001f000 fe:01 2504591 /lib/libz.so.1.2.13
fffff979a8000-fffff979a9000 rw-p 00020000 fe:01 2504591 /lib/libz.so.1.2.13
fffff979a9000-fffff97f9a000 r-xp 00000000 fe:01 2508107 /usr/local/lib/libruby.so.3.2.2
fffff97f9a000-fffff97fb9000 r--p 005f1000 fe:01 2508107 /usr/local/lib/libruby.so.3.2.2
fffff97fb9000-fffff97fba000 rw-p 00610000 fe:01 2508107 /usr/local/lib/libruby.so.3.2.2
fffff97fba000-fffff97fcb000 rw-p 00000000 00:00 0
fffff97fcb000-fffff98069000 r-xp 00000000 fe:01 2504585 /lib/ld-musl-aarch64.so.1
fffff98069000-fffff98077000 rw-p 00000000 00:00 0
fffff98077000-fffff98079000 r--p 00000000 00:00 0 [vvar]
fffff98079000-fffff9807a000 r-xp 00000000 00:00 0 [vdso]
fffff9807a000-fffff9807c000 rw-p 0009f000 fe:01 2504585 /lib/ld-musl-aarch64.so.1
fffff9807c000-fffff9807e000 rw-p 00000000 00:00 0
fffff980f0000-fffff980f3000 r-xp 00000000 00:00 0
fffff980f3000-fffff9c0f0000 ---p 00000000 00:00 0
fffffef20c000-fffffefa0b000 rw-p 00000000 00:00 0 [stack]

```

#3 - 08/03/2023 08:14 AM - willnet (Shinichi Maeshima)

□□□□□□□□□□□□

- Docker version 24.0.2, build cb74dfc
- M1 Max
- Ventura 13.5

#4 - 08/23/2023 10:16 PM - alanwu (Alan Wu)

- Status changed from Open to Closed

Applied in changeset [git|2214bcb70d9f9120f1f3790ca340236c8f080991](https://gitlab.com/containers/docker/-/commit/2214bcb70d9f9120f1f3790ca340236c8f080991).

Fix premature string collection during append

Previously, the following crashed due to use-after-free with AArch64 Alpine Linux 3.18.3 (aarch64-linux-musl):

```
str = 'a' * (32*1024*1024)
p({z: str})
```

32 MiB is the default for GC_MALLOC_LIMIT_MAX, and the crash could be dodged by setting RUBY_GC_MALLOC_LIMIT_MAX to large values. Under a debugger, one can see the str2 of rb_str_buf_append() getting prematurely collected while str_buf_cat4() allocates capacity.

Add GC guards so the buffer of str2 lives across the GC run initiated in str_buf_cat4().

[Bug [#19792](#)]

#5 - 08/23/2023 10:27 PM - alanwu (Alan Wu)

- Backport changed from 3.0: UNKNOWN, 3.1: UNKNOWN, 3.2: UNKNOWN to 3.0: UNKNOWN, 3.1: UNKNOWN, 3.2: REQUIRED

aaaaaaaaaaaaaaaa

#6 - 09/30/2023 07:09 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 3.0: UNKNOWN, 3.1: UNKNOWN, 3.2: REQUIRED to 3.0: UNKNOWN, 3.1: UNKNOWN, 3.2: DONE

ruby_3_2 d30781db4de82a891712f359d7659c9fc98cb215 merged revision(s) 2214bcb70d9f9120f1f3790ca340236c8f080991.

#7 - 03/27/2024 12:48 AM - hsbt (Hiroshi SHIBATA)

- Tags set to musl, alpine