Ruby trunk - Bug #5801

Enumerable#take_while □ proc □□□□□□□□□□ Segv

12/23/2011 05:31 PM - shiba (satoshi shiba)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 1.9.3p6 (2011-12-20 revision 34080) [i686-linux]	**Backport:**

Description

□□□□□□□

□□□□□□□□□□□□□Enumerable#take_while □ Proc □
□□□□□□□□□ Segv □□□□□□□

```
# □□□□□
class A
  include Enumerable
  def each(&b)
    $block = b
    yield
  end
end
puts A.new.take_while{true}
100.times{$block.call}
```

$block □□□take_while □□□□□□ take_while_i (enum.c)
□ C □ Proc □□□□□□□□□□ Proc □□□□□□□□
□□□□□□□□□□□□□□□take_while □□□□□□□□
□□□ Segv □□□□□□

□□□□□□□□enum_take_while □ take_while_i □□
take_while □□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□

```
/* enum_take_while □ take_while_i */
static VALUE
enum_take_while(VALUE obj)
{
  VALUE ary;
  RETURN_ENUMERATOR(obj, 0, 0);
  ary = rb_ary_new();
  rb_block_call(obj, id_each, 0, 0, take_while_i, (VALUE)&ary /* <= □□ */);
  return ary;
}

static VALUE
take_while_i(VALUE i, VALUE *ary /* <= □□ */, int argc, VALUE *argv)
{
  if (!RTEST(enum_yield(argc, argv))) rb_iter_break();
  rb_ary_push(*ary /* <= □□ */, enum_values_pack(argc, argv));
  return Qnil;
}
```

enum_take_while □□ rb_block_call □□□□□□□□ &ary □
□□□□□□□□□□□□□□ary □ C □□□□□□□□□□□□□□
C □□□□□□□□□□□□□□□□□□□□□□□□take_while □
□□□□□□□□□□□□□□□□□□□□□□ Proc □□ take_while_i
□□□□□□□□□□*ary □□□□□□□□□□□□□□□□ Segv □
□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□

```
/* □□□ */
Index: enum.c
===================================================================
--- enum.c      (revision 34107)
+++ enum.c      (working copy)
@@ -2053,7 +2053,7 @@
 take_while_i(VALUE i, VALUE *ary, int argc, VALUE *argv)
 {
     if (!RTEST(enum_yield(argc, argv))) rb_iter_break();
-    rb_ary_push(*ary, enum_values_pack(argc, argv));
+    rb_ary_push((VALUE)ary, enum_values_pack(argc, argv));
     return Qnil;
 }

@@ -2079,7 +2079,7 @@

     RETURN_ENUMERATOR(obj, 0, 0);
     ary = rb_ary_new();
-    rb_block_call(obj, id_each, 0, 0, take_while_i, (VALUE)&ary);
+    rb_block_call(obj, id_each, 0, 0, take_while_i, ary);
     return ary;
 }
```

Associated revisions

Revision 41e331cc - 02/17/2012 04:24 AM - nobu (Nobuyoshi Nakada)

- enum.c: move work variables to objects not to let called blocks access stack area out of scope. [Bug #5801]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@34661 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 34661 - 02/17/2012 04:24 AM - nobu (Nobuyoshi Nakada)

- enum.c: move work variables to objects not to let called blocks access stack area out of scope. [Bug #5801]

Revision 34661 - 02/17/2012 04:24 AM - nobu (Nobuyoshi Nakada)

- enum.c: move work variables to objects not to let called blocks access stack area out of scope. [Bug #5801]

Revision 34661 - 02/17/2012 04:24 AM - nobu (Nobuyoshi Nakada)

- enum.c: move work variables to objects not to let called blocks access stack area out of scope. [Bug #5801]

Revision 34661 - 02/17/2012 04:24 AM - nobu (Nobuyoshi Nakada)

- enum.c: move work variables to objects not to let called blocks access stack area out of scope. [Bug #5801]

Revision 34661 - 02/17/2012 04:24 AM - nobu (Nobuyoshi Nakada)

- enum.c: move work variables to objects not to let called blocks access stack area out of scope. [Bug #5801]

History

#1 - 12/23/2011 05:45 PM - shiba (satoshi shiba)

Enumerable#grep □□□□□□□□□□□□□□□□
□□□□□□□□□□ Segv □□□□□□

enum.c □□XX_i □□□□□□□□□□□□□ C □
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□

□ □ □ □ □

```
class A
  include Enumerable
  def each(&b)
    $block = b
    yield
  end
end
puts A.new.grep(//)
100.times{$block.call}

static VALUE
grep_i(VALUE i, VALUE args, int argc, VALUE *argv)
{
    VALUE *arg = (VALUE *)args; /* arg □ C □□□□□□□□ */
    ENUM_WANT_SVALUE();

    if (RTEST(rb_funcall(arg[0], id_eqq, 1, i))) {
        rb_ary_push(arg[1], i);
    }
    return Qnil;
}
```

#2 - 02/17/2012 01:24 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r34661.
satoshi, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- enum.c: move work variables to objects not to let called blocks access stack area out of scope. [Bug #5801]

#3 - 06/12/2015 12:38 AM - nobu (Nobuyoshi Nakada)

- Description updated

#4 - 06/12/2015 12:40 AM - nobu (Nobuyoshi Nakada)

- Description updated