Ruby master - Bug #18475

Yielding an element for Enumerator in another thread dumps core

01/11/2022 06:24 PM - mame (Yusuke Endoh)

Status: Closed
Priority: Normal
Assignee: Target version:
ruby -v:
Backport: 2.6: UNKNOWN, 2.7: REQUIRED, 3.0: DONE, 3.1: DONE

Description
I faced [BUG] when investigating #18474

def run
  Thread.new do
    1.times do |value|
      yield "some-value"
    end
  end.join
end
to_enum(:run).first

$ ruby repro.rb
<Thread:0x00007fcab26af0c0 repro.rb:2 run> terminated with exception (report_on_exception is true):
[BUG] Segmentation fault at 0x0000000000000018

-- Control frame information ----------------------------------------------
c:0001 p:----- s:0003 e:000002 (none) [FINISH]

-- Machine register context -----------------------------------------------
RIP: 0x0000561c86b90c25 RBP: 0x0000000000000000 RSP: 0x000007fcab259d970
RAX: 0x0000000000000000 RBX: 0x000007fcab259d9f8 RCX: 0x0000000000002021
RDI: 0x0000000000000000 RSI: 0x000000000000009b
R8: 0x0000000000000001 R9: 0x0000000000000000 R10: 0x0000000000000000
R11: 0x0000000000000293 R12: 0x0000000000000000 R13: 0x000007fcab26a660
R14: 0x0000000000002021 R15: 0x000000000000009b EFL: 0x0000000000010202

-- C level backtrace information ------------------------------------------
./home/mame/local/bin/ruby(rb_print_backtrace+0x11) [0x561c86bb541d] vm_dump.c:759
./home/mame/local/bin/ruby(rb_vm_bugreport) vm_dump.c:1045
./home/mame/local/bin/ruby(sigsegv+0x4d) [0x561c86b0898d] signal.c:964
/lib/x86_64-linux-gnu/libpthread.so.0(__restore_rt+0x0) [0x7fcab64ff1f0] ../sysdeps/pthread/funloc
kfile.c:28
./home/mame/local/bin/ruby(cached_callable_method_entry+0x0) [0x561c86b90c25] vm_sync.h:40
./home/mame/local/bin/ruby(cached_callable_method_entry) vm_method.c:1316
./home/mame/local/bin/ruby(vm_response_to+0x3f) [0x561c86ba099f] vm_method.c:2693
./home/mame/local/bin/ruby(check_funcall_response_to+0xb) [0x561c86ba137c] vm_eval.c:603
./home/mame/local/bin/ruby(rb_check_funcall_default_kw) vm_eval.c:677
./home/mame/local/bin/ruby(rb_get_message+0x18) [0x561c86c7ea98] error.c:1231
./home/mame/local/bin/ruby(rb_ec_error_print+0x1c3) [0x561c869a9983] eval_error.c:378
./home/mame/local/bin/ruby(thread_start_func+0x6d7) [0x561c86b50737] thread.c:847
./home/mame/local/bin/ruby(thread_start_func+0xe0) [0x561c86b90c25] vm_sync.h:40
./home/mame/local/bin/ruby(thread_start_func+0xe0) [0x561c86b50737] thread.c:847
./home/mame/local/bin/ruby(thread_start_func_and_wait) thread_pthread.c:1099
./home/mame/local/bin/ruby(thread_start_func_and_wait) thread_pthread.c:1099
/lib/x86_64-linux-gnu/libpthread.so.0(start+0xe0) [0x7fcab64f4450] pthread_create.c:473
/lib/x86_64-linux-gnu/libpthread.so.0(start+0xe0) [0x7fcab627ad53] .. /sysdeps/unix/sysv/linux/x86_64/clo
Related issues:
Related to Ruby master - Feature #17613: Eliminate useless catch tables and nops from lambdas Closed
Related to Ruby master - Bug #11254: [BUG] no corresponding cfp Closed
Related to Ruby master - Bug #18474: 938e027c seems to have caused a regression... Open

Associated revisions
Revision ae5458f2 - 01/12/2022 06:21 PM - mame (Yusuke Endoh)
thread.c: Convert TAG_BREAK to a normal exception at thread top-level
[Bug #18475]

Revision 8e3bce08 - 01/31/2022 02:16 AM - naruse (Yui NARUSE)
merge revision(s) ae5458f228a5477383e9c00425d85d50a3867817: [Backport #18475]

Revision 42b1e87b - 03/13/2022 02:37 AM - nagachika (Tomoyuki Chikanaga)
merge revision(s) ae5458f228a5477383e9c00425d85d50a3867817: [Backport #18475]

History
#1 - 01/12/2022 02:36 AM - mame (Yusuke Endoh)
- Related to Feature #17613: Eliminate useless catch tables and nops from lambdas added

#2 - 01/12/2022 07:27 AM - mame (Yusuke Endoh)
- Related to Bug #11254: [BUG] no corresponding cfp added

#3 - 01/12/2022 07:27 AM - mame (Yusuke Endoh)
- Related to Bug #18474: 938e027c seems to have caused a regression in yield handling with concurrent-ruby added

#4 - 01/12/2022 08:31 AM - mame (Yusuke Endoh)
This is almost the same issue as #11254. When TAG_BREAK is thrown to the top-level frame, it dumps core. #11254 fixed the case where TAG_BREAK is thrown to the top-level of the main thread, but not of a non-main thread.

I think this patch will fix the issue. It converts uncaught TAG_BREAK (or other non-local jumps) to a normal exception at the top-level of a thread.

diff --git a/thread.c b/thread.c
index 36723c6fc0..cf6e89327e 100644
--- a/thread.c
+++ b/thread.c
@@ -830,6 +830,9 @@ thread_start_func_2(rb_thread_t *th, VALUE *stack_start)
 else {
     errinfo = th->ec->errinfo;
 +     VALUE exc = rb_vm_make_jump_tag_but_local_jump(state, Qundef);
 +     if (!NIL_P(exc)) errinfo = exc;
 +     if (state == TAG_FATAL) {
         if (th->invoke_type == thread_invoke_type_ractor_proc) {
             rb_ractor_atexit(th->ec, Qnil);
     
03/15/2022
thread.c: Convert TAG_BREAK to a normal exception at thread top-level

[Bug #18475]

#7 - 01/12/2022 06:22 PM - mame (Yusuke Endoh)
- Backport changed from 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN to 2.6: UNKNOWN, 2.7: REQUIRED, 3.0: REQUIRED, 3.1: REQUIRED

#8 - 01/31/2022 02:17 AM - naruse (Yui NARUSE)
- Backport changed from 2.6: UNKNOWN, 2.7: REQUIRED, 3.0: REQUIRED, 3.1: REQUIRED to 2.6: UNKNOWN, 2.7: REQUIRED, 3.0: REQUIRED, 3.1: DONE

ruby_3_1 8e3bce0841497fc00ccdaf45f6583c25720a2d3f merged revision(s) ae5458f228a5477383e9c00425d85d50a3867817.

#9 - 03/13/2022 03:12 AM - nagachika (Tomoyuki Chikanaga)
- Backport changed from 2.6: UNKNOWN, 2.7: REQUIRED, 3.0: REQUIRED, 3.1: DONE to 2.6: UNKNOWN, 2.7: REQUIRED, 3.0: DONE, 3.1: DONE

ruby_3_0 42b1e87ba7856f29297392485a63a60b5fa750c8 merged revision(s) ae5458f228a5477383e9c00425d85d50a3867817.