# Ruby - Bug #20942

## Infinite loop when out of memory

12/11/2024 02:05 AM - segiddins (Samuel Giddins)

| | | | |
|---|---|---|---|
| **Status:** | Closed | | |
| **Priority:** | Normal | | |
| **Assignee:** | | | |
| **Target version:** | | | |
| **ruby -v:** | ruby 3.4.0dev (2024-12-10T10:28:22Z master 3568e7aef7) +PRISM [aarch64-linux] | **Backport:** | 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN |

**Description**

Similar setup to https://bugs.ruby-lang.org/issues/20941
Pure-ruby reproduction of https://bugs.ruby-lang.org/issues/20629

```
#!/usr/bin/env -S RUBYOPT="--disable-gems"  gdb --args ruby -v

STDOUT.sync = true

es = Array.new(10000) { Object.new }
es = Array.new(10000) { 0 }

Process.warmup
puts "warmup"

Process.setrlimit(:DATA, 0)

i = 0
loop do
  begin
    Array.new(-1)
  rescue ArgumentError => e
    es[i += 1] = e
  end
end
```

Backtrace every time I pause in the debugger:

```
(gdb) bt
#0  rb_multi_ractor_p () at /usr/src/ruby/vm_sync.h:40
#1  rb_vm_lock_leave (line=245, file=<synthetic pointer>, lev=<synthetic pointer>) at /usr/src/rub
y/vm_sync.h:92
#2  rb_ec_vm_lock_rec_release (ec=ec@entry=0xaaaaaaabb4f0, recorded_lock_rec=0, current_lock_rec=1
) at /usr/src/ruby/vm_sync.c:245
#3  0x0000fffff7c3badc in rb_ec_vm_lock_rec_check (recorded_lock_rec=<optimized out>, ec=0xaaaaaaa
bb4f0) at /usr/src/ruby/eval_intern.h:136
#4  rb_ec_tag_state (ec=0xaaaaaaabb4f0) at /usr/src/ruby/eval_intern.h:147
#5  rb_vm_exec (ec=0xaaaaaaabb4f0) at /usr/src/ruby/vm.c:2584
#6  0x0000fffff7a52c18 in rb_ec_exec_node (ec=ec@entry=0xaaaaaaabb4f0, n=n@entry=0xfffff72c5068) a
t /usr/src/ruby/eval.c:281
#7  0x0000fffff7a563b4 in ruby_run_node (n=0xfffff72c5068) at /usr/src/ruby/eval.c:319
#8  0x0000aaaaaaaa0b6c in rb_main (argv=0xffffffffffce8, argc=3) at /usr/src/ruby/main.c:43
#9  main (argc=<optimized out>, argv=<optimized out>) at /usr/src/ruby/main.c:68
```

### Associated revisions

**Revision 2f6c694977dc0cdc4766a8a921e74290963c19a7 - 12/20/2024 07:49 AM - eightbitraptor (Matt V-H)**

Memerror is fatal if VM cannot be unlocked.

[Bug #20942]

If we've raised a memerror while the VM is locked, and the tag we're

jumping to has been locked at a different level to the current lock (ie. we've locked the VM again since the tag we're jumping to) then we should consider this memerror fatal and exit, since the tag cannot unlock the VM.

Co-Authored-By: Peter Zhu peter@peterzhu.ca

**Revision 2f6c694977dc0cdc4766a8a921e74290963c19a7 - 12/20/2024 07:49 AM - eightbitraptor (Matt V-H)**

Memerror is fatal if VM cannot be unlocked.

[Bug #20942]

If we've raised a memerror while the VM is locked, and the tag we're jumping to has been locked at a different level to the current lock (ie. we've locked the VM again since the tag we're jumping to) then we should consider this memerror fatal and exit, since the tag cannot unlock the VM.

Co-Authored-By: Peter Zhu peter@peterzhu.ca

**Revision 2f6c6949 - 12/20/2024 07:49 AM - eightbitraptor (Matt V-H)**

Memerror is fatal if VM cannot be unlocked.

[Bug #20942]

If we've raised a memerror while the VM is locked, and the tag we're jumping to has been locked at a different level to the current lock (ie. we've locked the VM again since the tag we're jumping to) then we should consider this memerror fatal and exit, since the tag cannot unlock the VM.

Co-Authored-By: Peter Zhu peter@peterzhu.ca

## History

**#1 - 12/20/2024 07:49 AM - eightbitraptor (Matt V-H)**

*- Status changed from Open to Closed*

Applied in changeset git|2f6c694977dc0cdc4766a8a921e74290963c19a7.

---

Memerror is fatal if VM cannot be unlocked.

[Bug #20942]

If we've raised a memerror while the VM is locked, and the tag we're jumping to has been locked at a different level to the current lock (ie. we've locked the VM again since the tag we're jumping to) then we should consider this memerror fatal and exit, since the tag cannot unlock the VM.

Co-Authored-By: Peter Zhu peter@peterzhu.ca