Ruby Stability at Scale

Peter Zhu

Ruby Core Committer
Staff Developer, Shopify

blog.peterzhu.ca/assets/rails_world_2025_slides.pdf

Peter Zhu

- Based in Toronto, Canada
- Ruby Core Committer
- Staff Developer on the Ruby Infrastructure team at Shopify
- Co-author of Variable Width Allocation, `RUBY_FREE_AT_EXIT` and Modular GC in Ruby
- Author of `ruby_memcheck` and `autotuner`
- Photography geek, follow me @peterzhu.photos on Instagram!

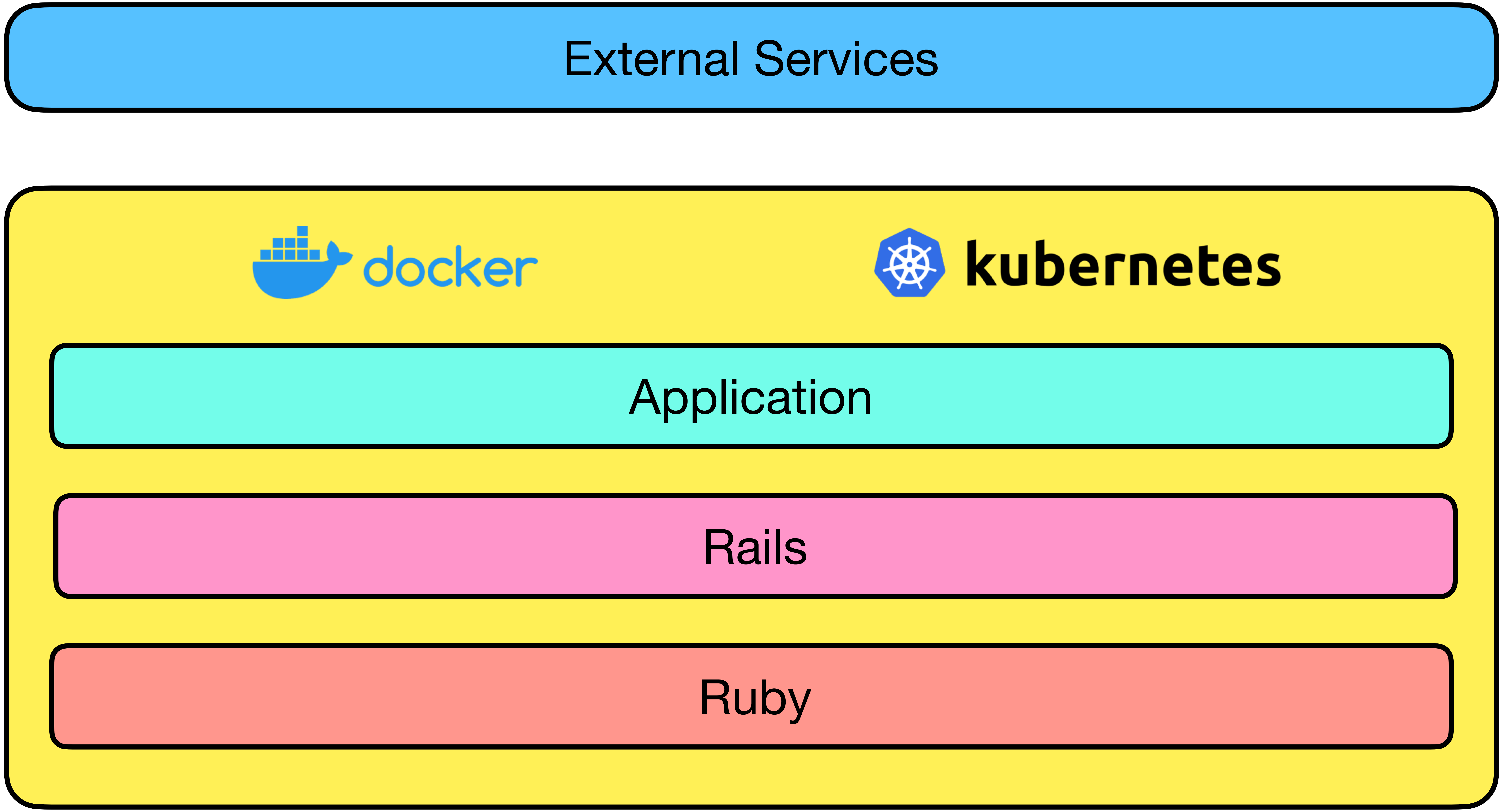
⛈️ Sources of Instability in Your Infrastructure

🐞 Sources of Instability in Ruby

🛡️ Preventing Crashes in Production

📉 Capturing Information About Crashes

Instability in Your Infrastructure



External Services

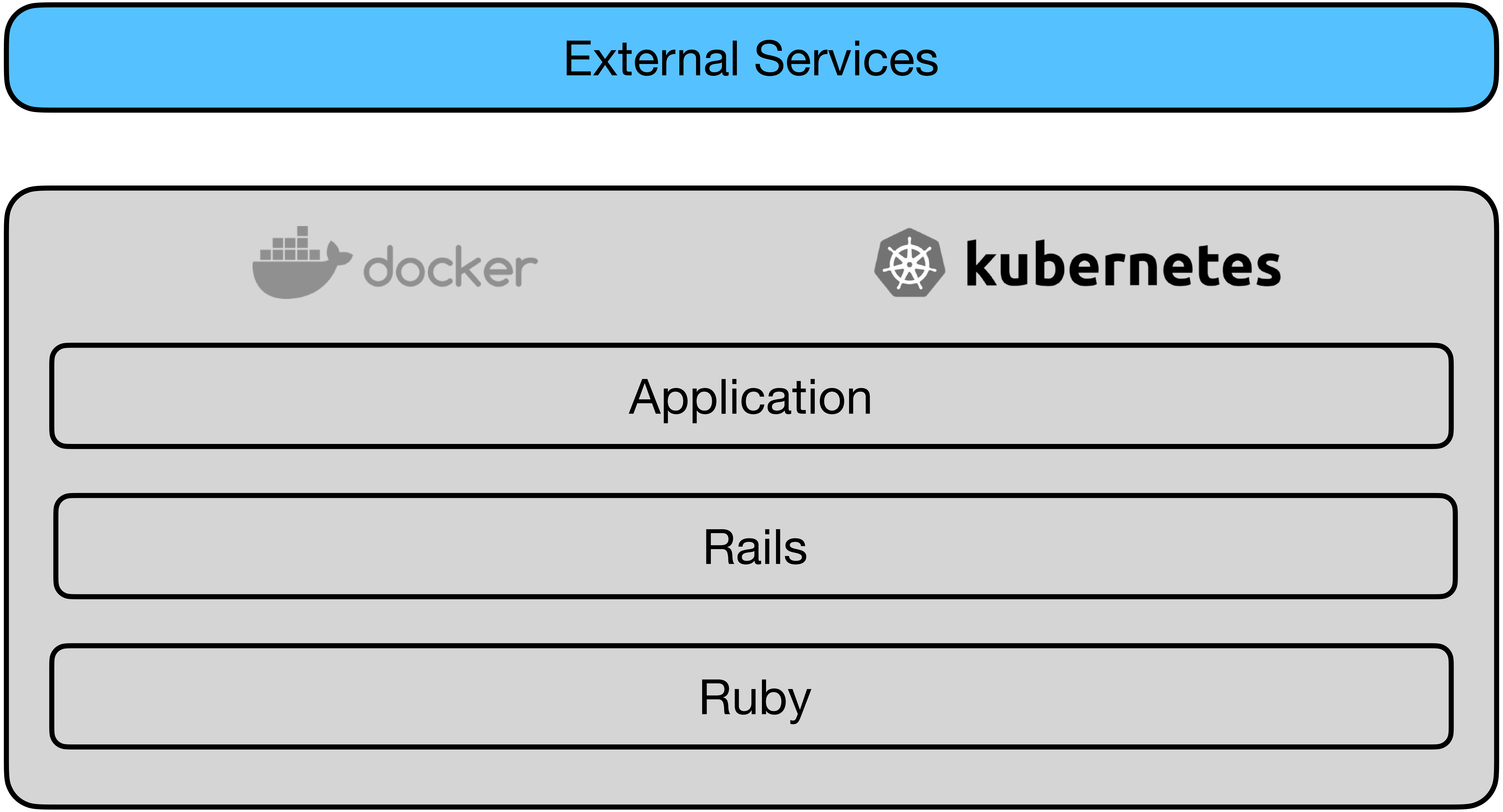
docker

kubernetes

Application

Rails

Ruby

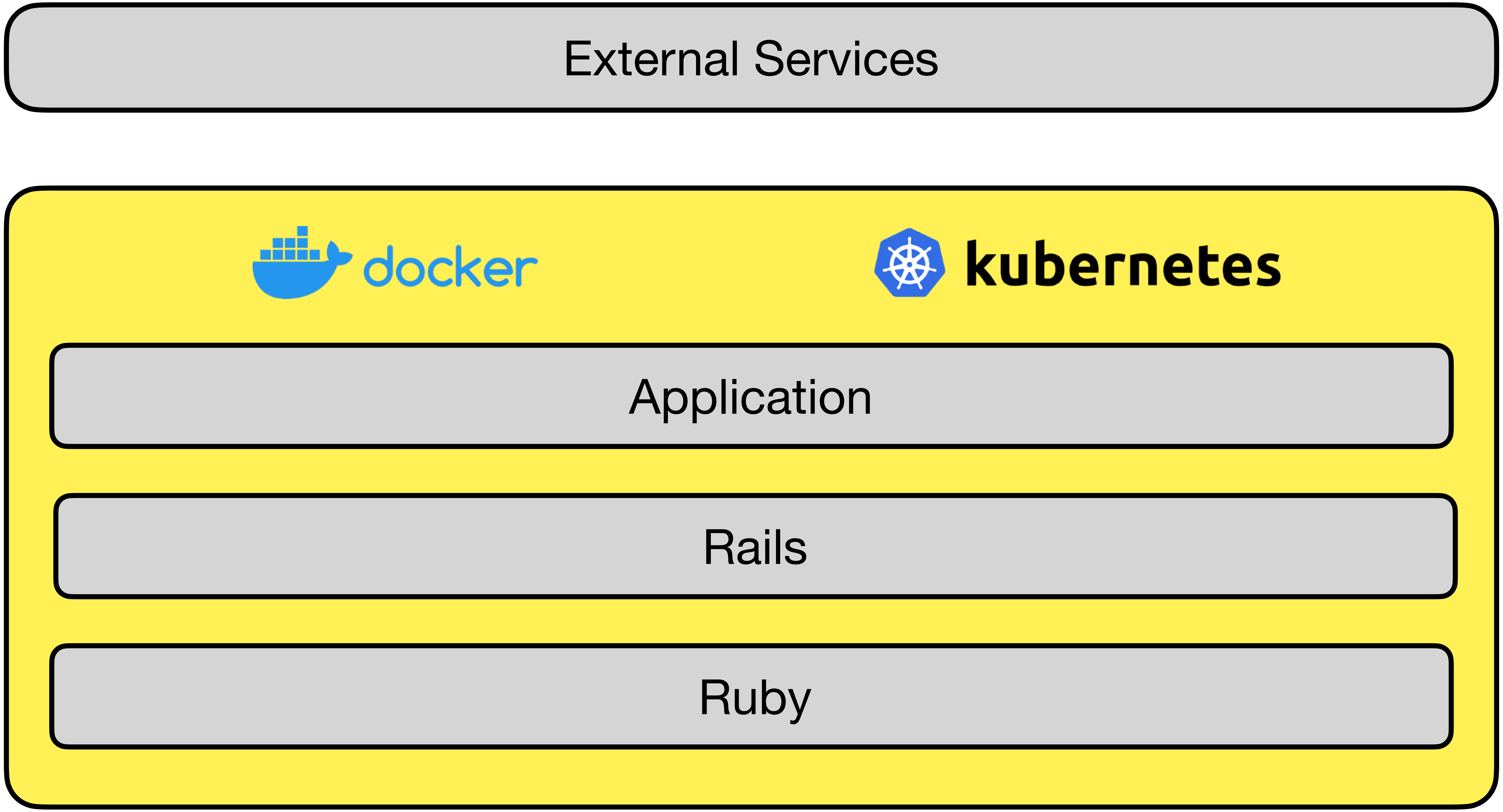


External Services

Application

Rails

Ruby

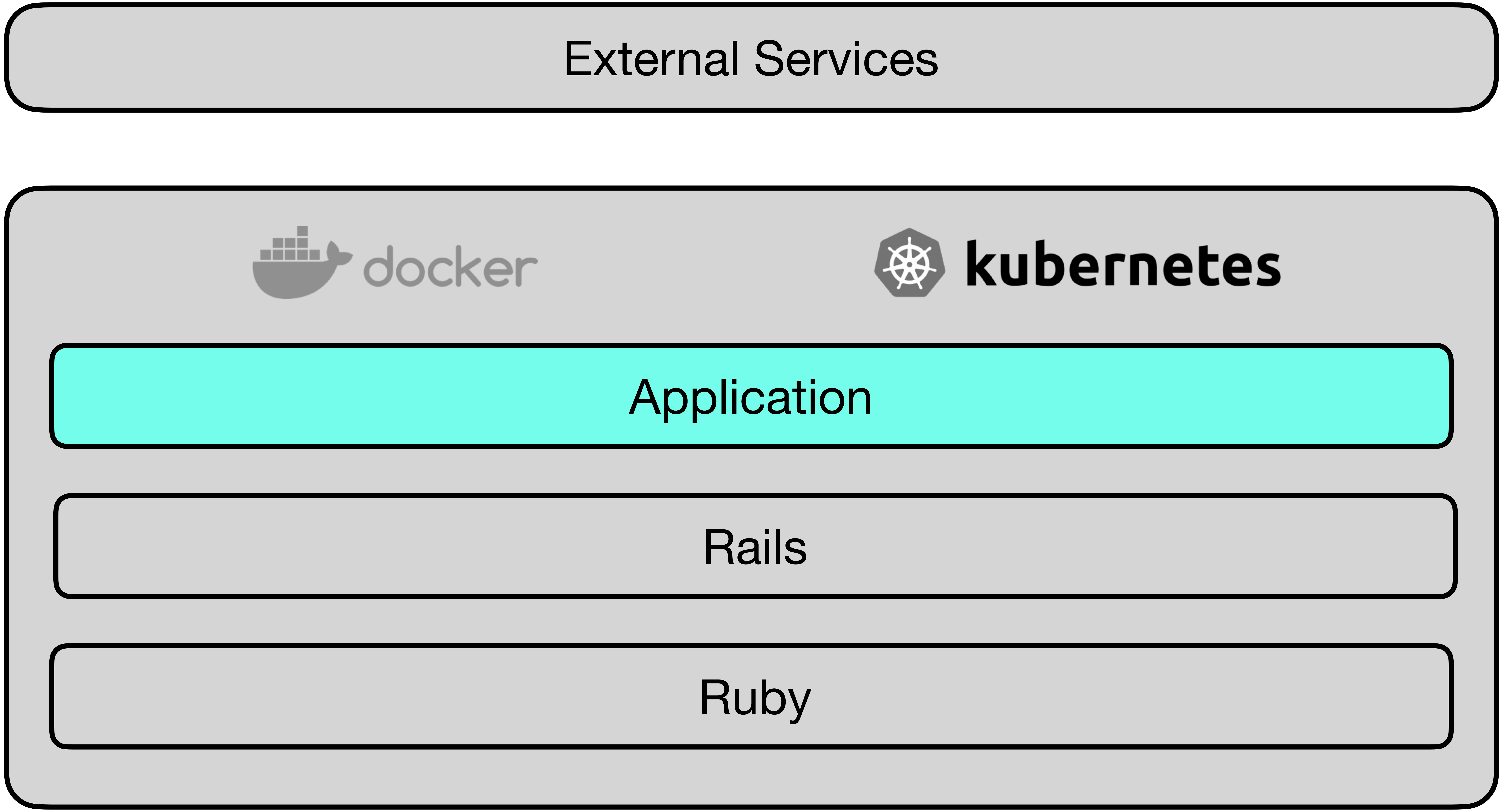


External Services

Application

Rails

Ruby



External Services

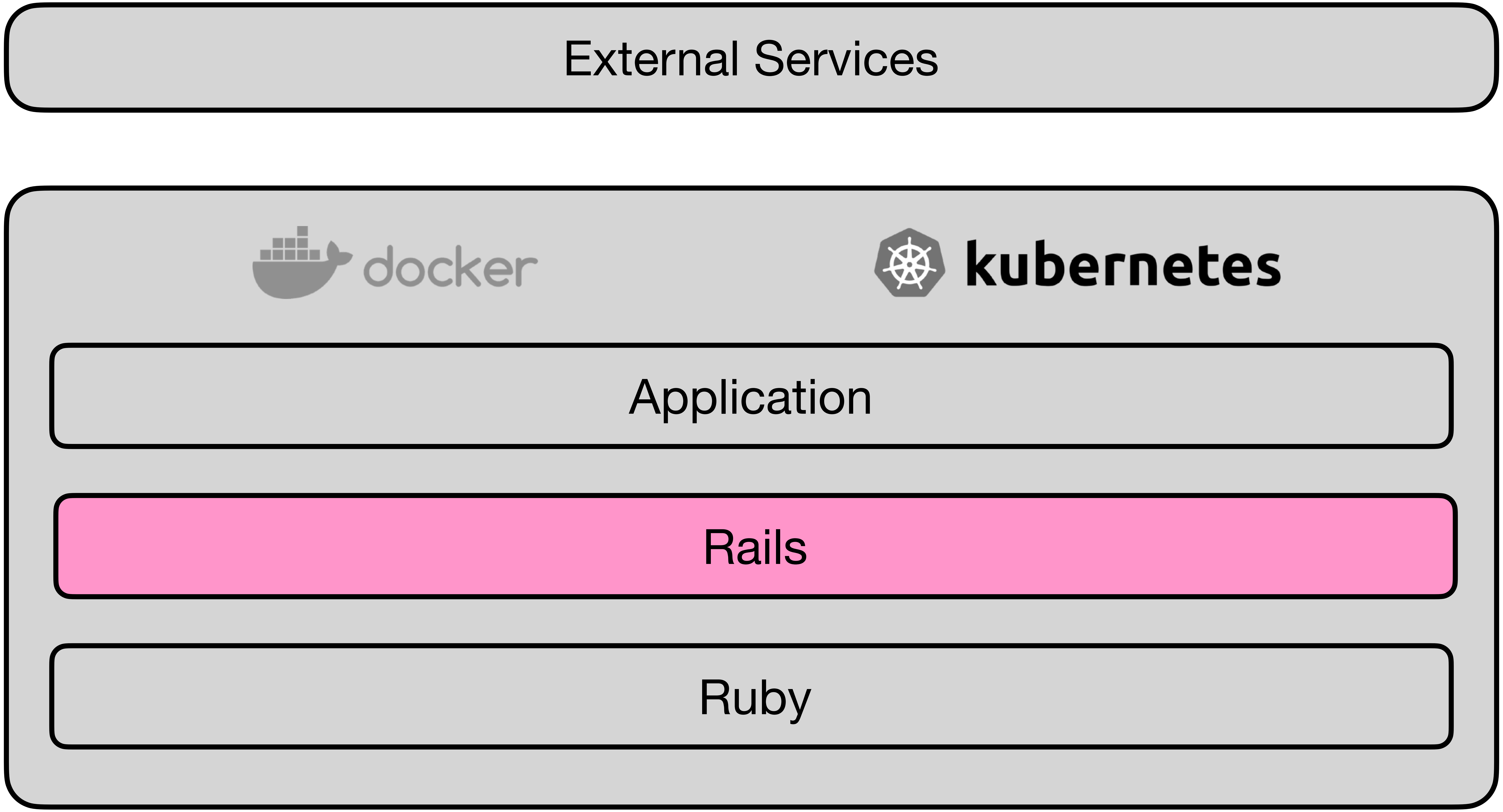
docker

kubernetes

Application

Rails

Ruby



External Services

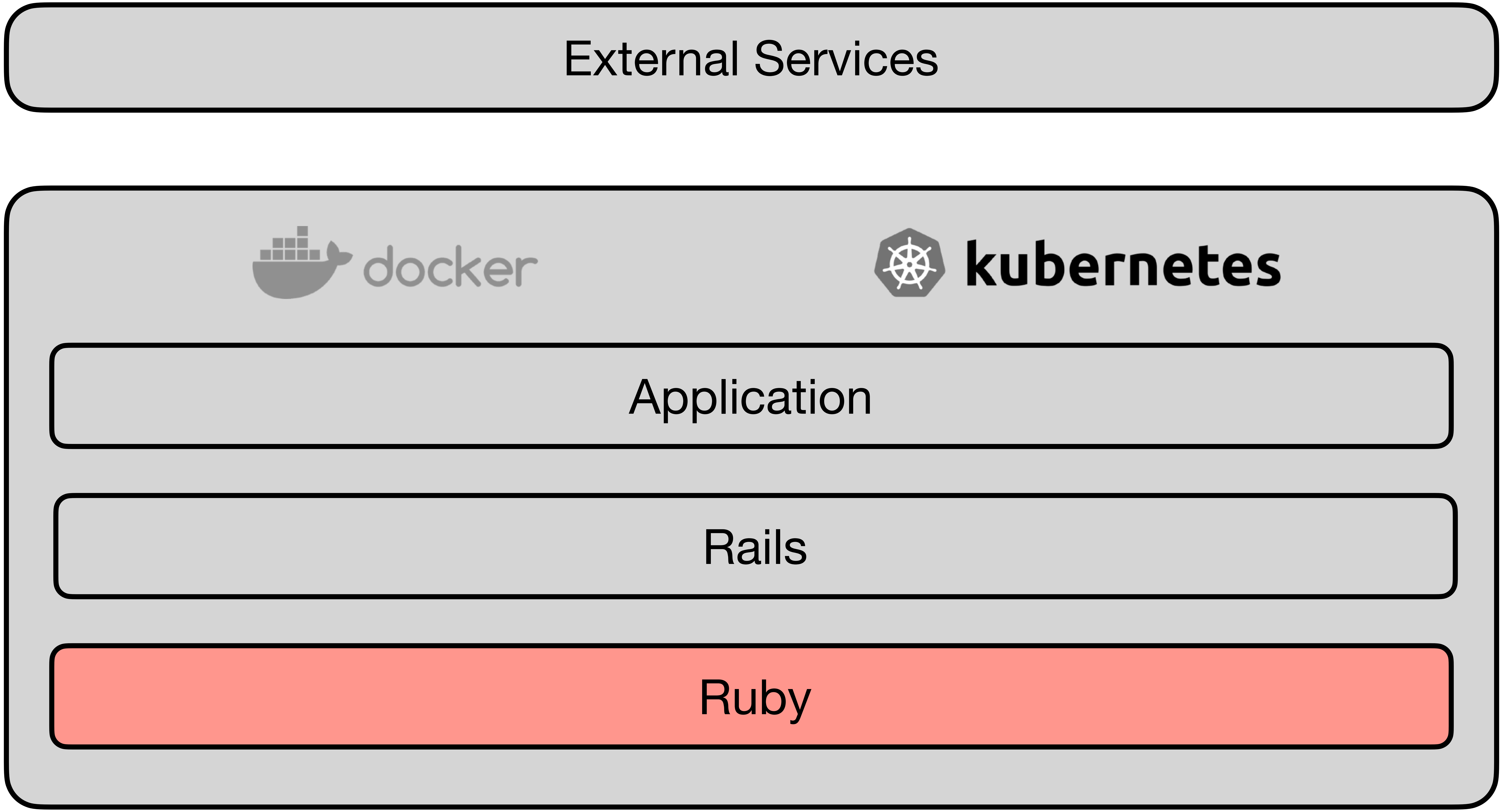
docker

kubernetes

Application

Rails

Ruby



External Services

docker

kubernetes

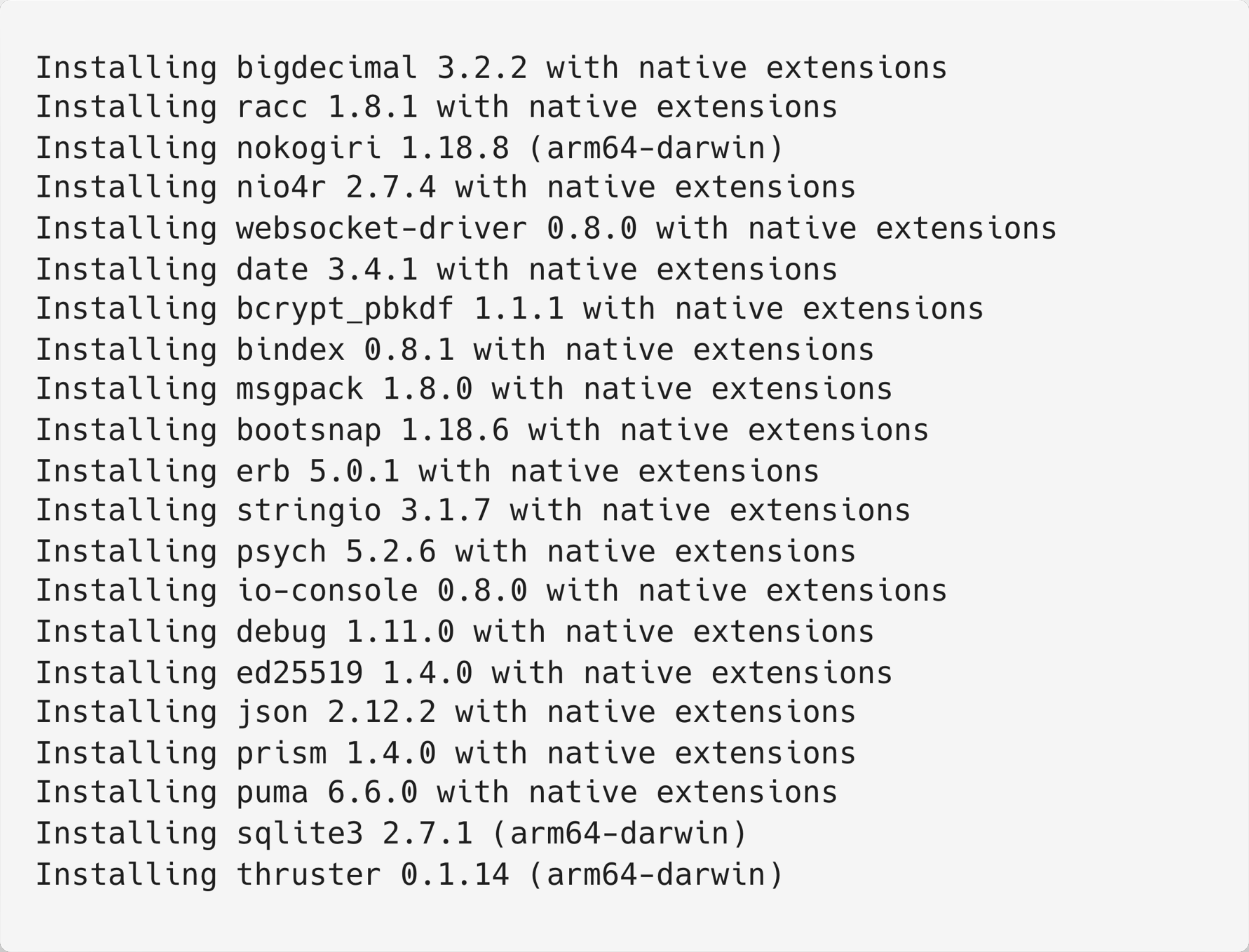
Application

Rails

Ruby

Sources of Instability in Ruby

Sources of Instability in Ruby & Native Gems



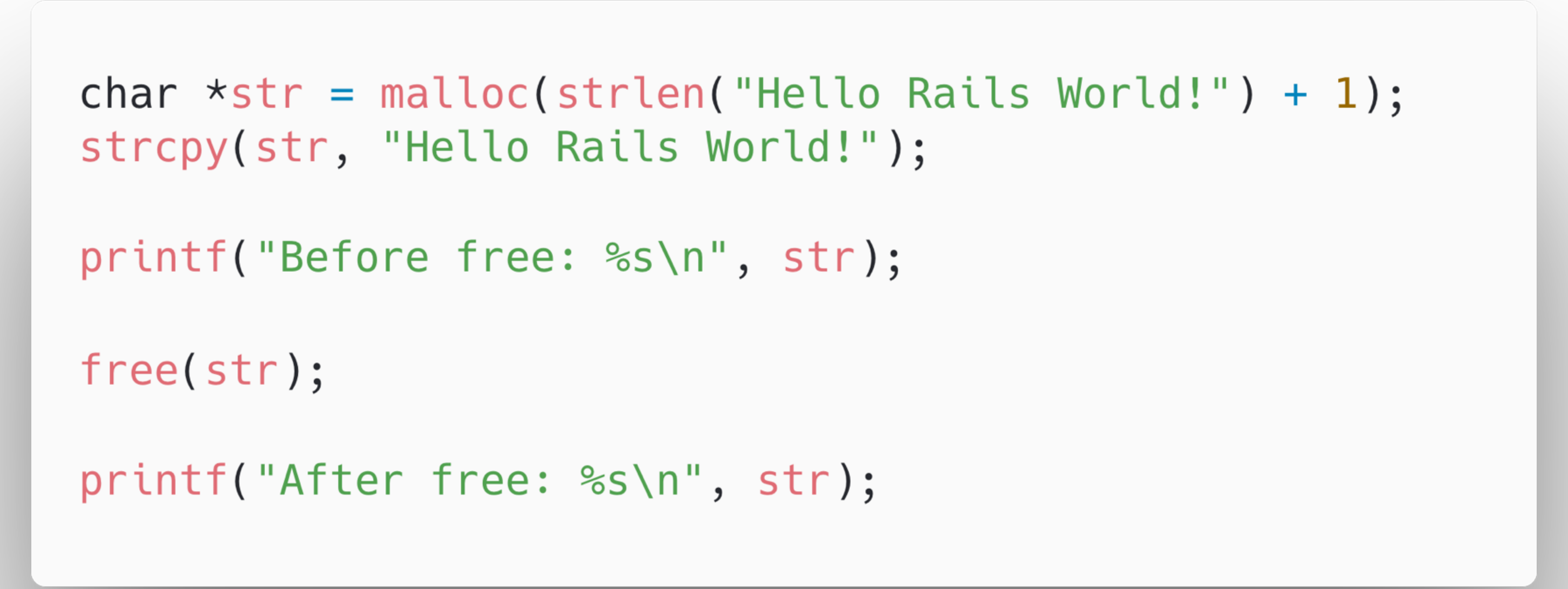
```
Installing bigdecimal 3.2.2 with native extensions
Installing racc 1.8.1 with native extensions
Installing nokogiri 1.18.8 (arm64-darwin)
Installing nio4r 2.7.4 with native extensions
Installing websocket-driver 0.8.0 with native extensions
Installing date 3.4.1 with native extensions
Installing bcrypt_pbkdf 1.1.1 with native extensions
Installing bindex 0.8.1 with native extensions
Installing msgpack 1.8.0 with native extensions
Installing bootsnap 1.18.6 with native extensions
Installing erb 5.0.1 with native extensions
Installing stringio 3.1.7 with native extensions
Installing psych 5.2.6 with native extensions
Installing io-console 0.8.0 with native extensions
Installing debug 1.11.0 with native extensions
Installing ed25519 1.4.0 with native extensions
Installing json 2.12.2 with native extensions
Installing prism 1.4.0 with native extensions
Installing puma 6.6.0 with native extensions
Installing sqlite3 2.7.1 (arm64-darwin)
Installing thruster 0.1.14 (arm64-darwin)
```

Sources of Instability in Ruby & Native Gems

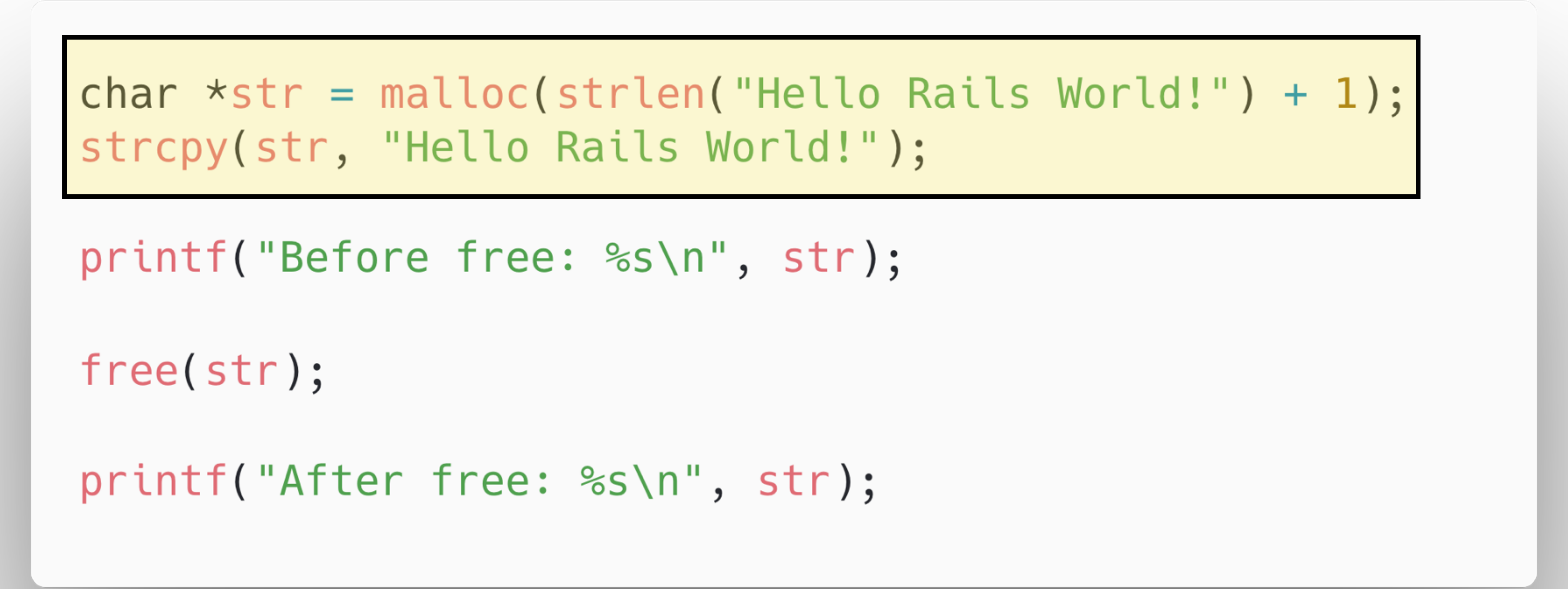
Bugs in C Code

Bugs in C Code

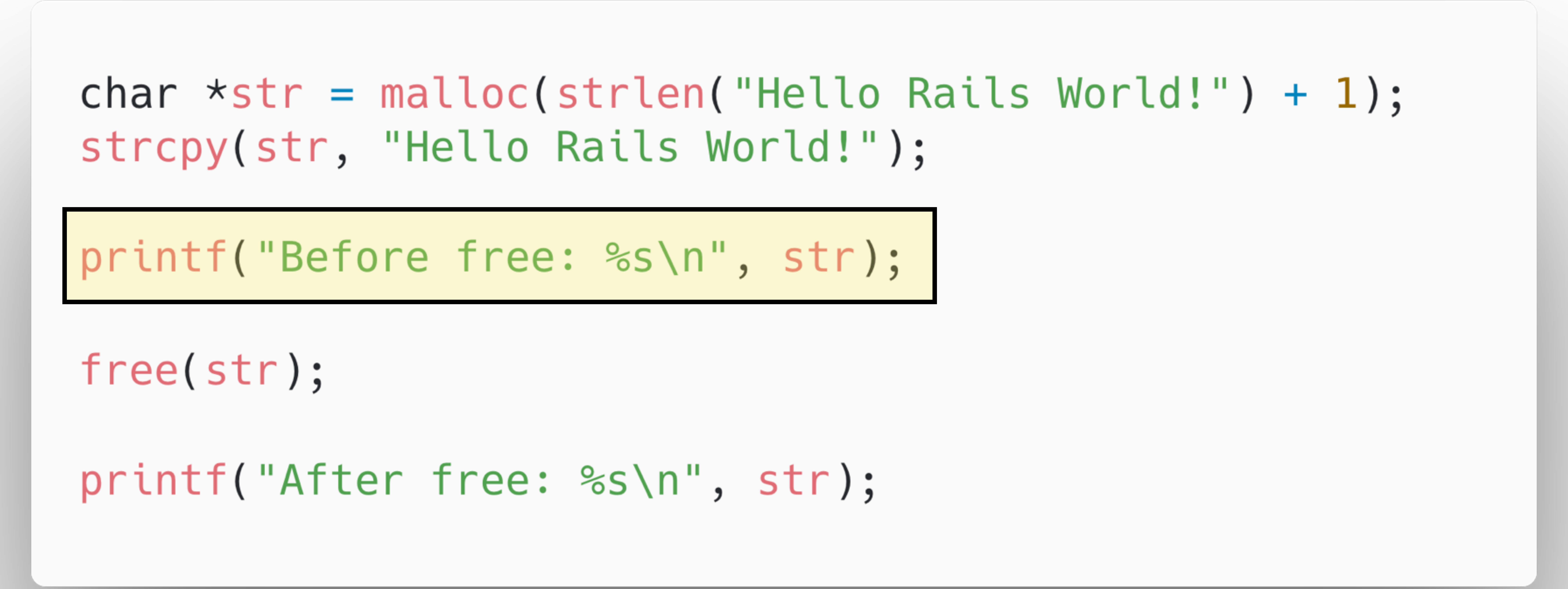
Use-After-Free



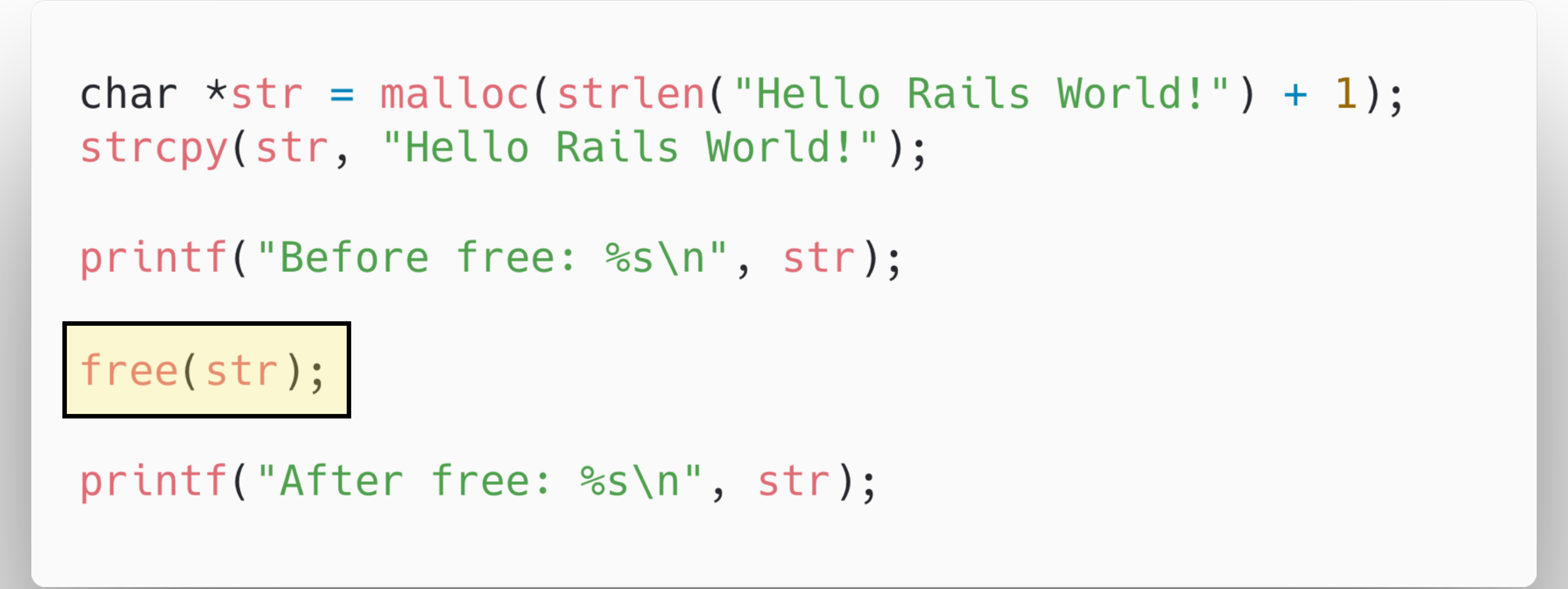
```c
char *str = malloc(strlen("Hello Rails World!") + 1);
strcpy(str, "Hello Rails World!");

printf("Before free: %s\n", str);

free(str);

printf("After free: %s\n", str);
```



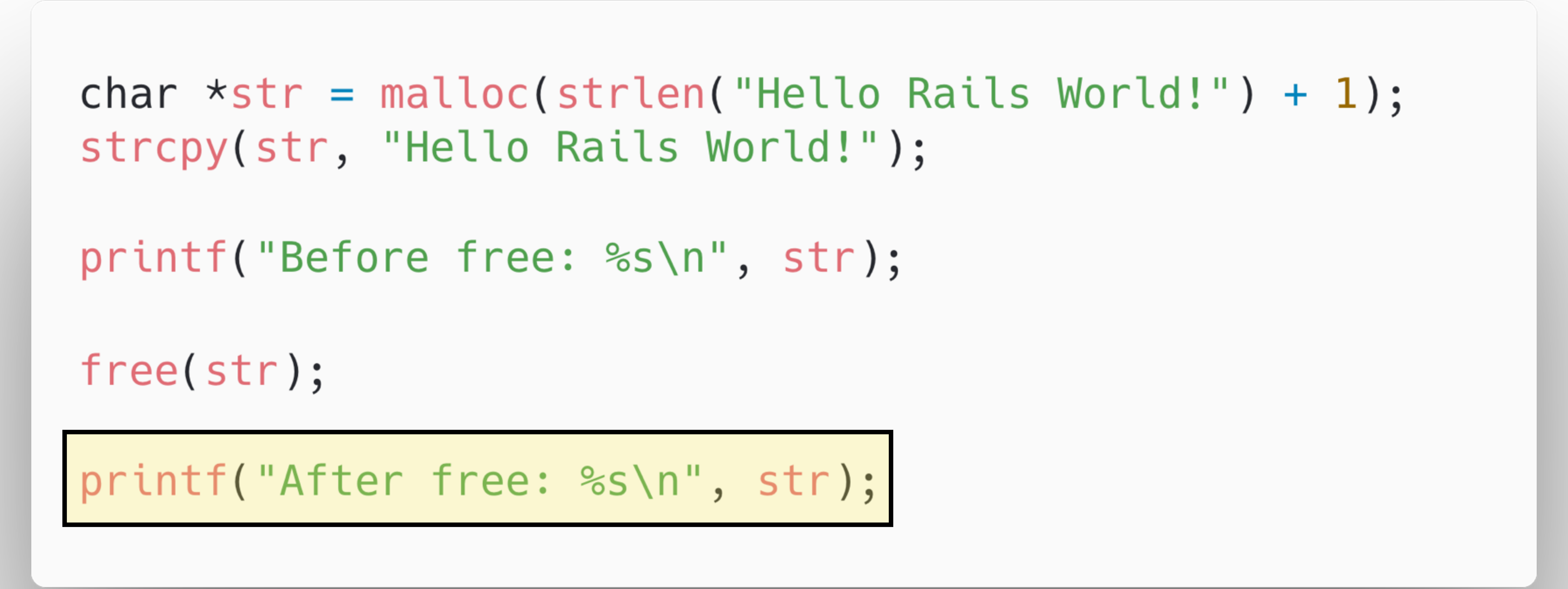
```
char *str = malloc(strlen("Hello Rails World!") + 1);
strcpy(str, "Hello Rails World!");

printf("Before free: %s\n", str);

free(str);

printf("After free: %s\n", str);
```



```c
char *str = malloc(strlen("Hello Rails World!") + 1);
strcpy(str, "Hello Rails World!");

printf("Before free: %s\n", str);

free(str);

printf("After free: %s\n", str);
```



```c
char *str = malloc(strlen("Hello Rails World!") + 1);
strcpy(str, "Hello Rails World!");

printf("Before free: %s\n", str);

free(str);

printf("After free: %s\n", str);
```



```c
char *str = malloc(strlen("Hello Rails World!") + 1);
strcpy(str, "Hello Rails World!");

printf("Before free: %s\n", str);

free(str);

printf("After free: %s\n", str);
```

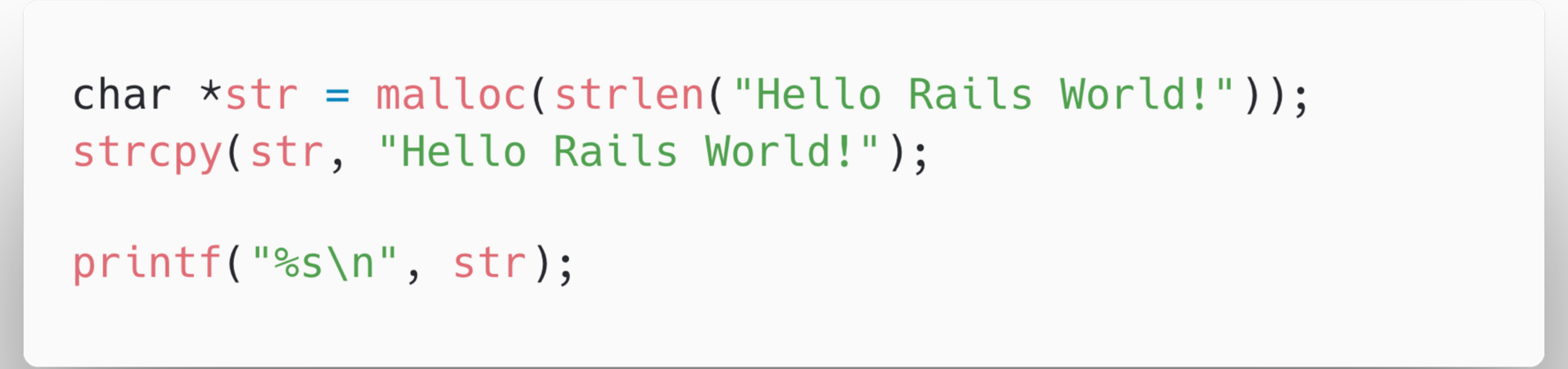
Bugs in C Code

Use-After-Free

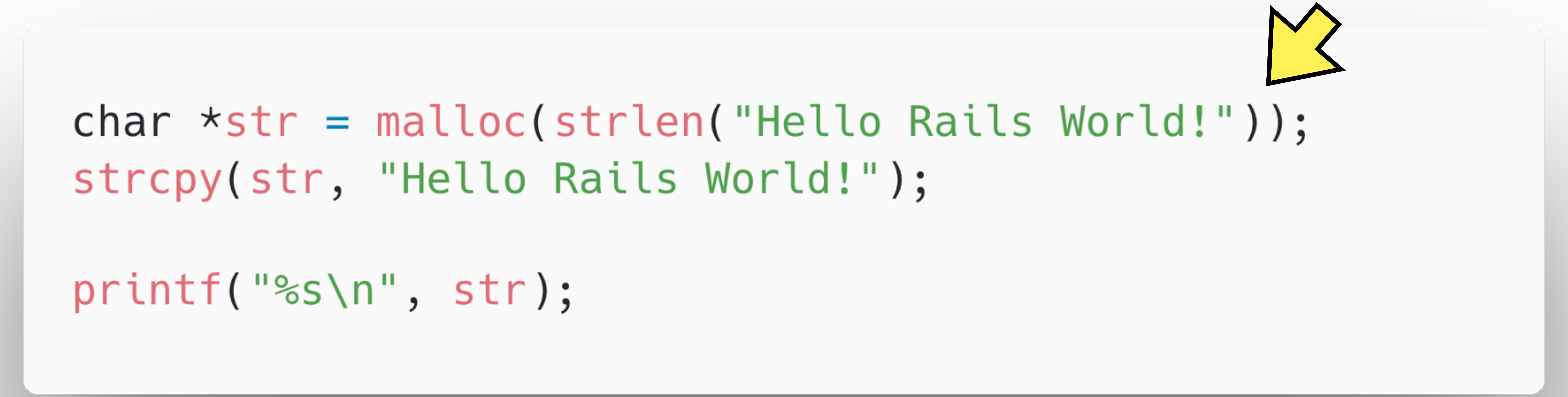
Bugs in C Code

Use-After-Free

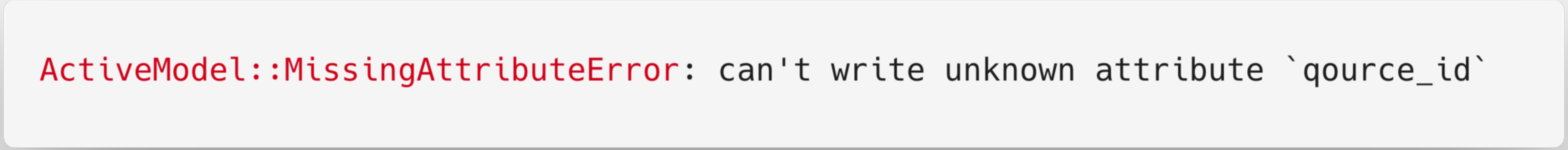
Buffer Overflow



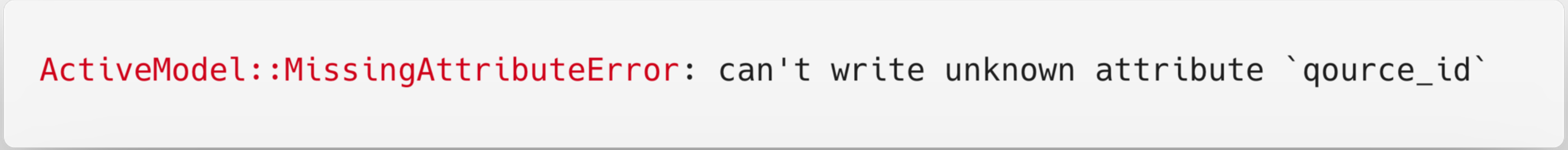
```c
char *str = malloc(strlen("Hello Rails World!"));
strcpy(str, "Hello Rails World!");

printf("%s\n", str);
```



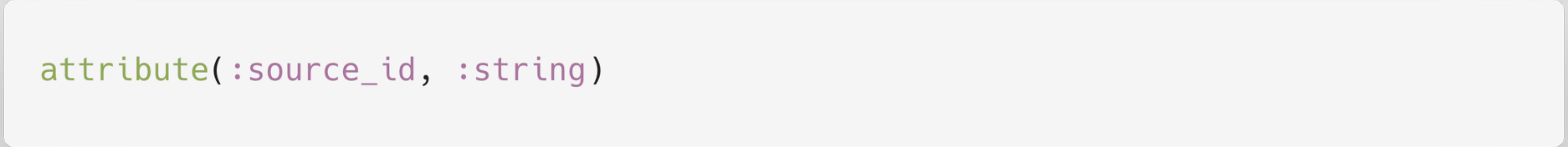
```
char *str = malloc(strlen("Hello Rails World!"));
strcpy(str, "Hello Rails World!");

printf("%s\n", str);
```



```
ActiveModel::MissingAttributeError: can't write unknown attribute `qource_id`
```



```
ActiveModel::MissingAttributeError: can't write unknown attribute `qource_id`
```



```
attribute(:source_id, :string)
```

s	73	01110011
q	71	01110001

Bugs in C Code

Use-After-Free

Buffer Overflow

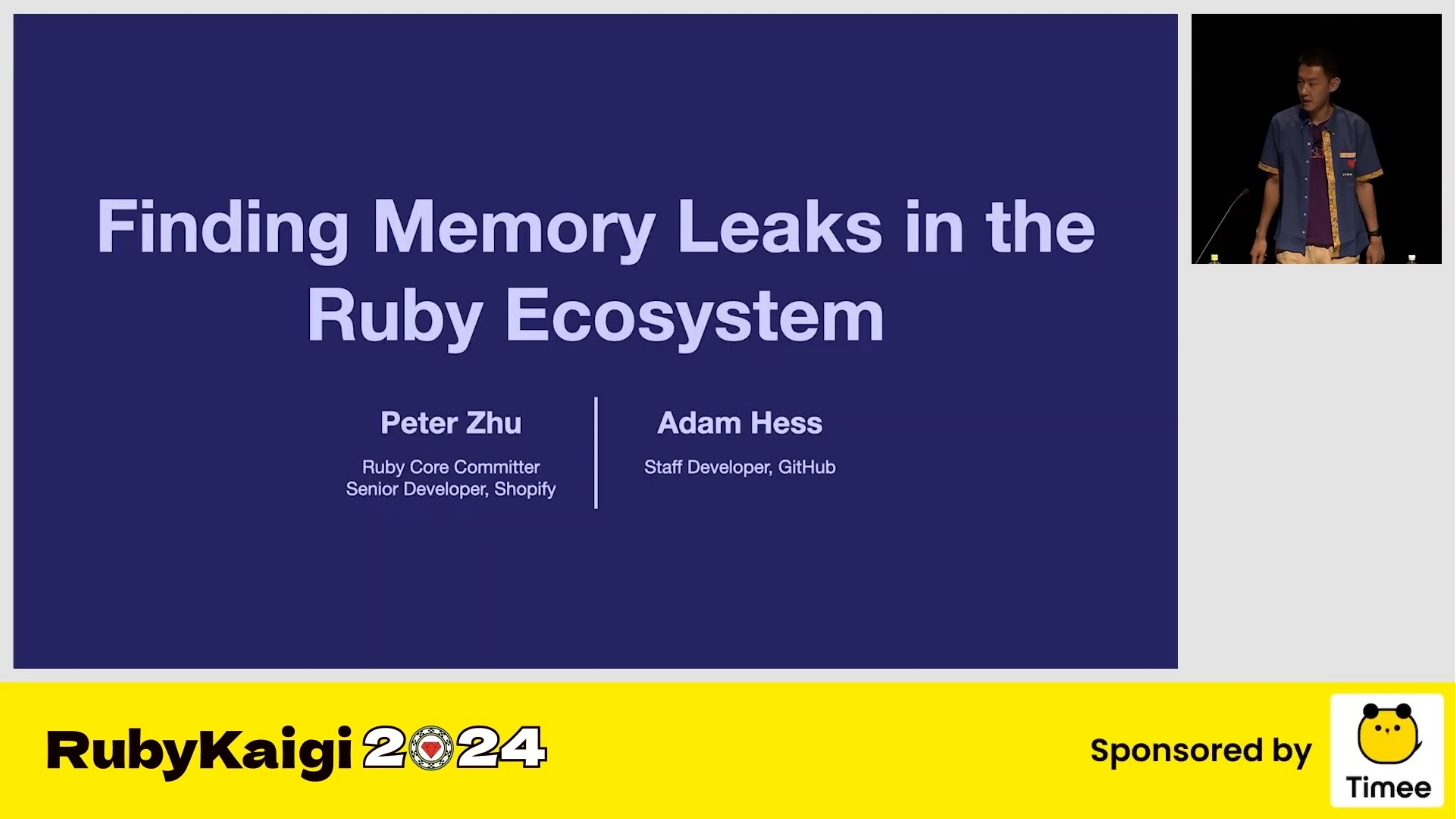
Bugs in C Code

Use-After-Free

Buffer Overflow

Memory Leak

youtu.be/pQ1XCrwq1qc



Finding Memory Leaks in the Ruby Ecosystem

Peter Zhu
Ruby Core Committer
Senior Developer, Shopify

Adam Hess
Staff Developer, GitHub

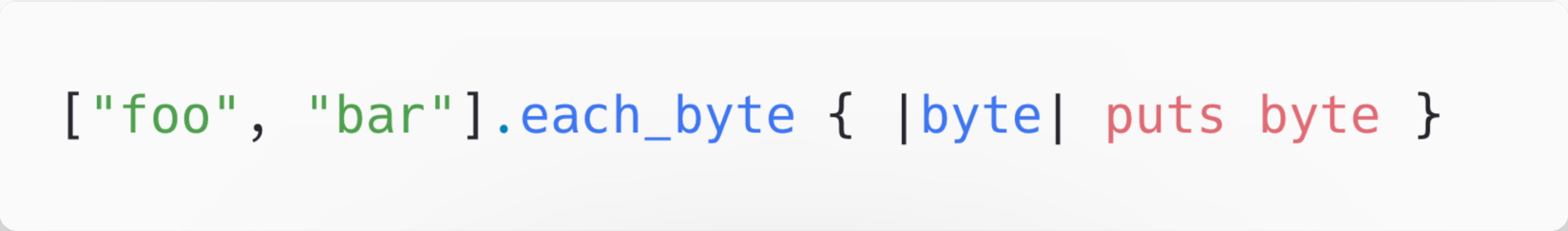
RubyKaigi 2024

Sponsored by Timee

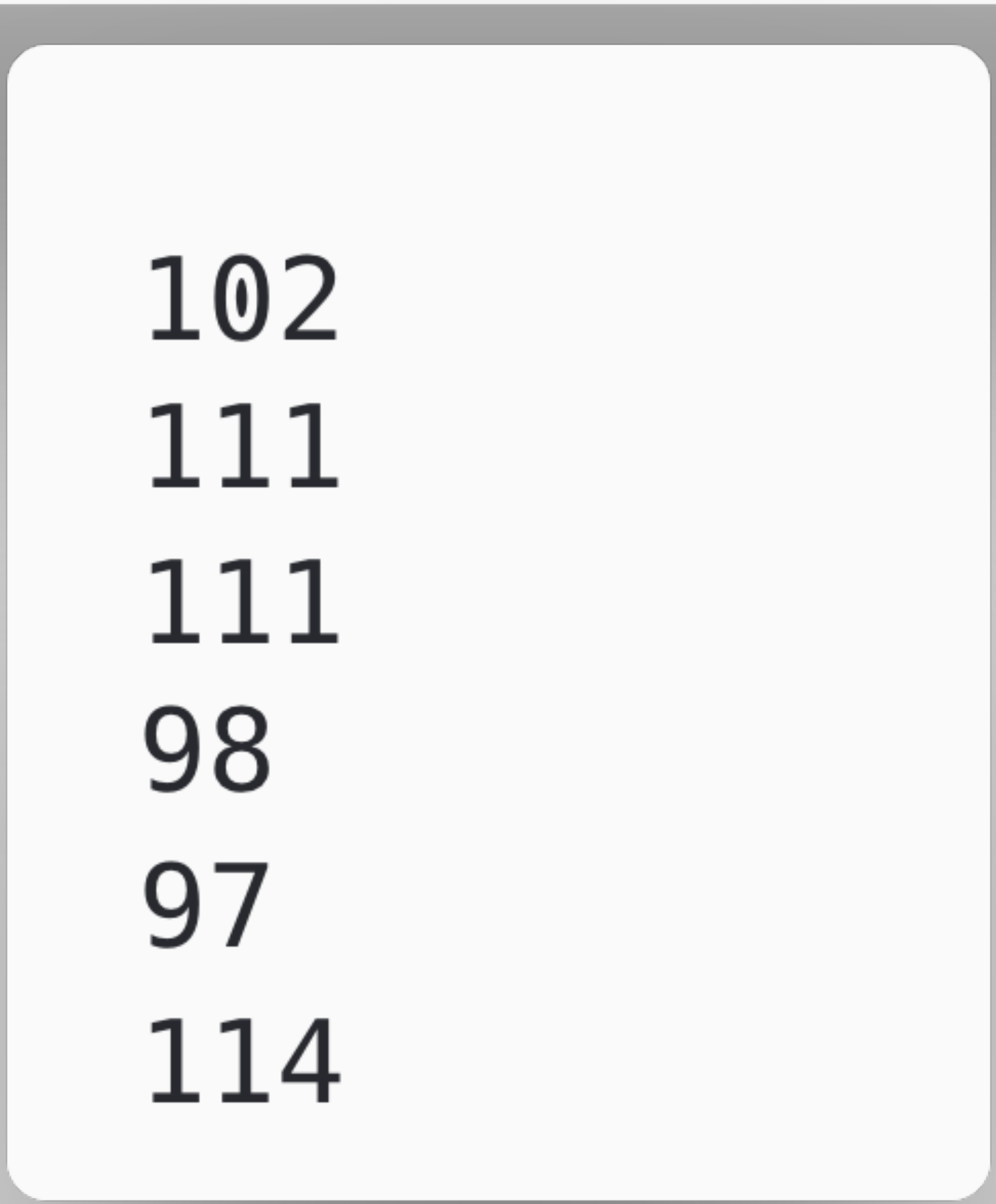
Incorrect Use of Ruby’s C API

Missing GC Guard

Array#each_byte

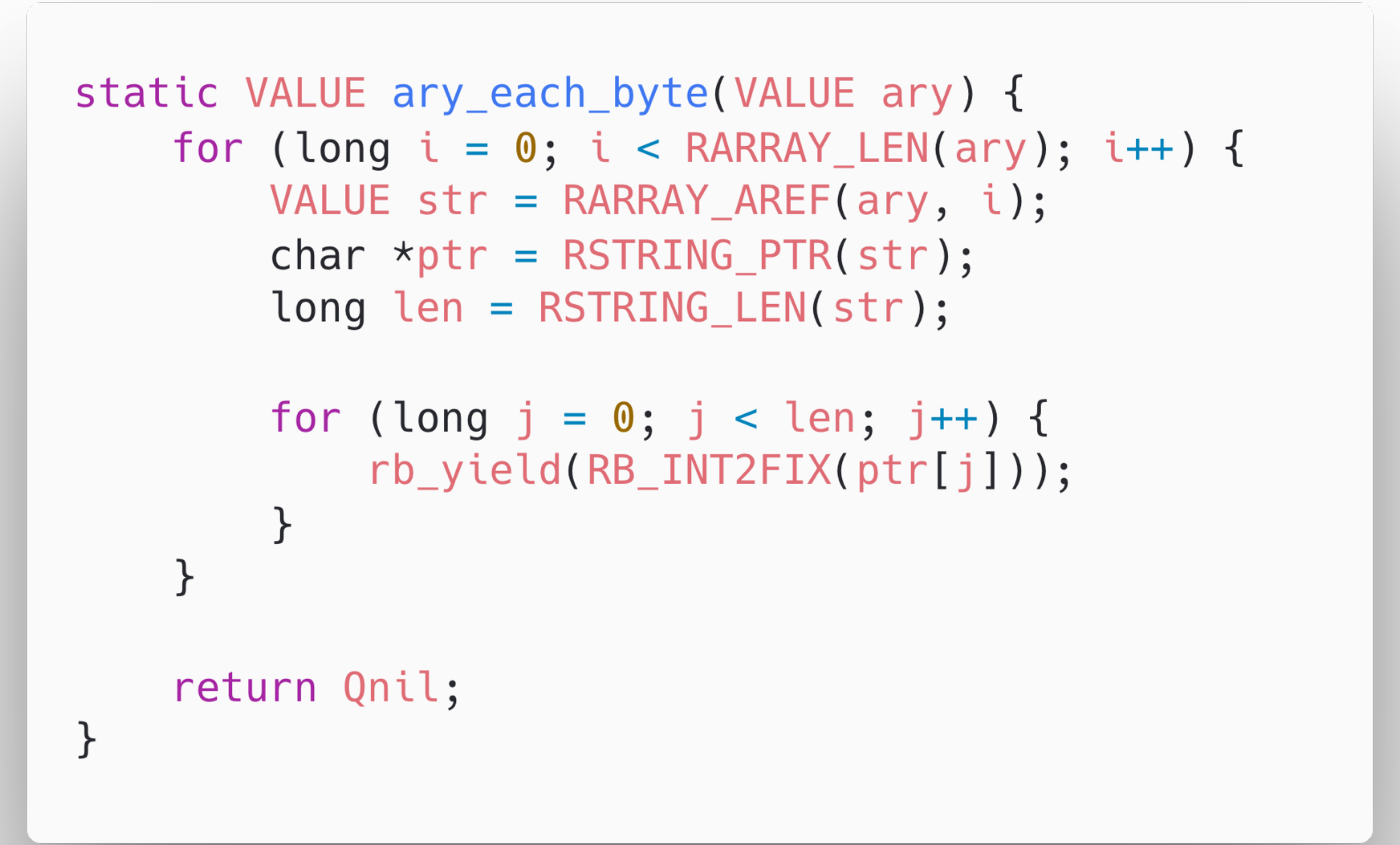


```
["foo", "bar"].each_byte { |byte| puts byte }
```



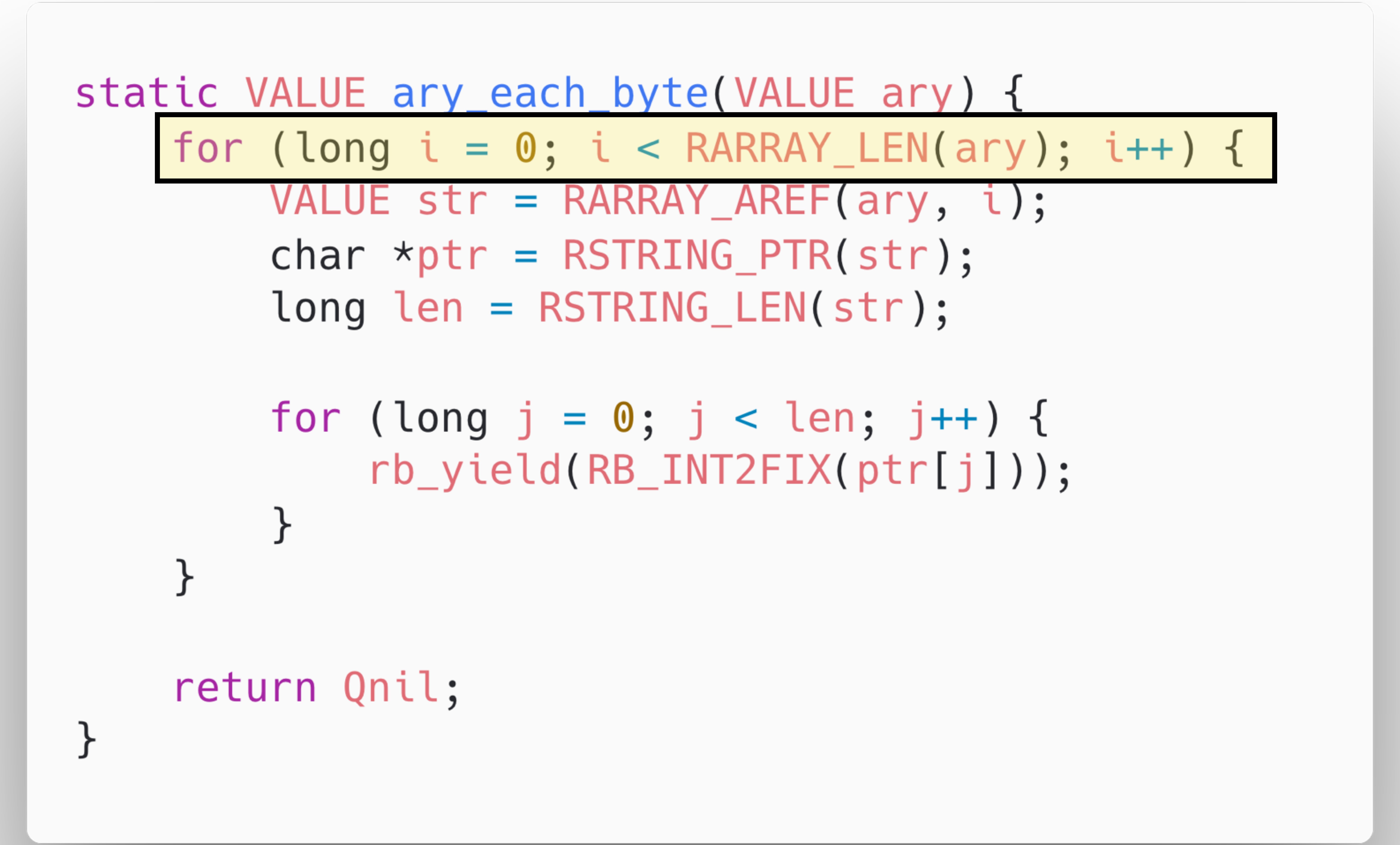
```
102
111
111
98
97
114
```

Array#each_byte



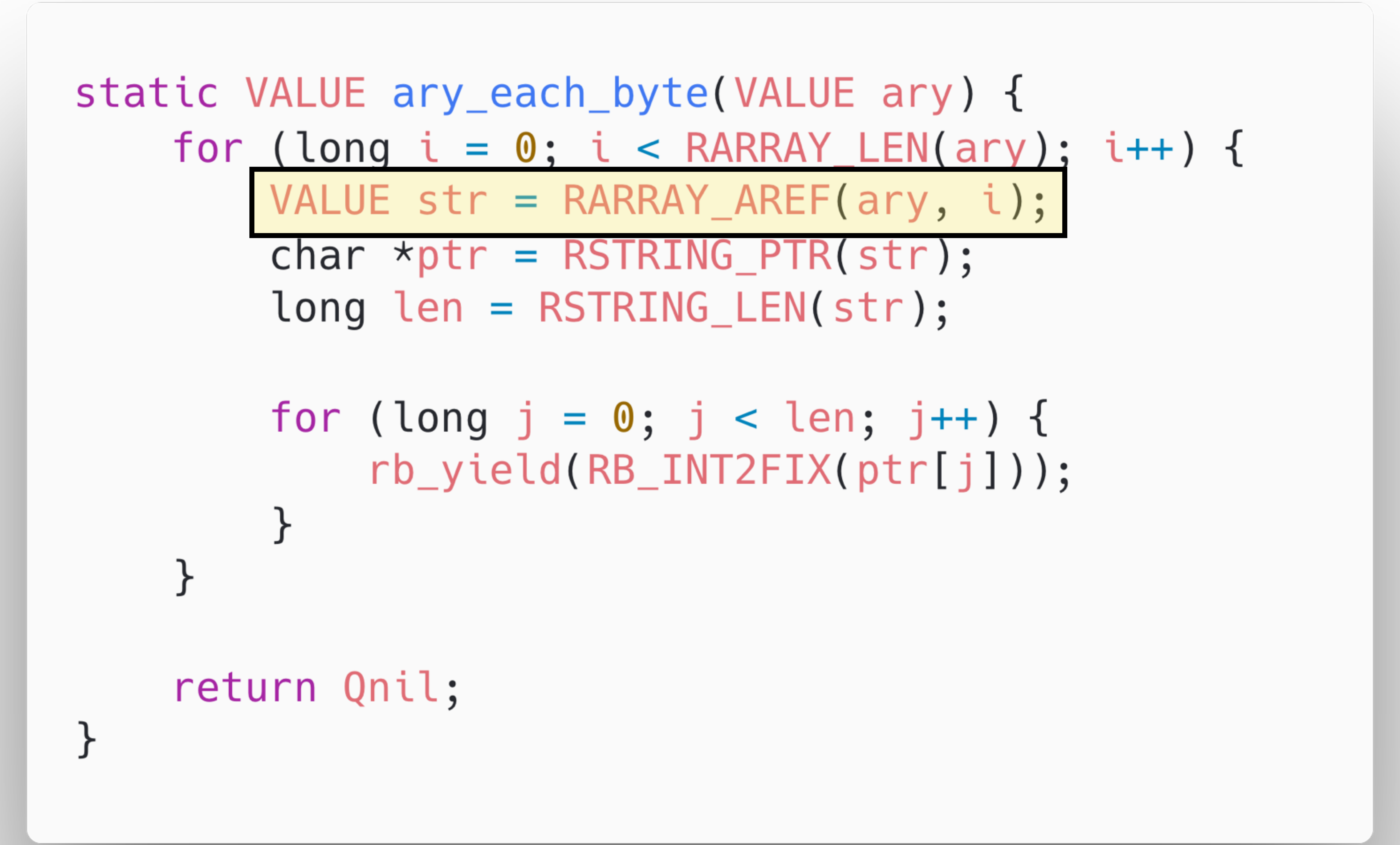
```c
static VALUE ary_each_byte(VALUE ary) {
    for (long i = 0; i < RARRAY_LEN(ary); i++) {
        VALUE str = RARRAY_AREF(ary, i);
        char *ptr = RSTRING_PTR(str);
        long len = RSTRING_LEN(str);

        for (long j = 0; j < len; j++) {
            rb_yield(RB_INT2FIX(ptr[j]));
        }
    }

    return Qnil;
}
```

Array#each_byte



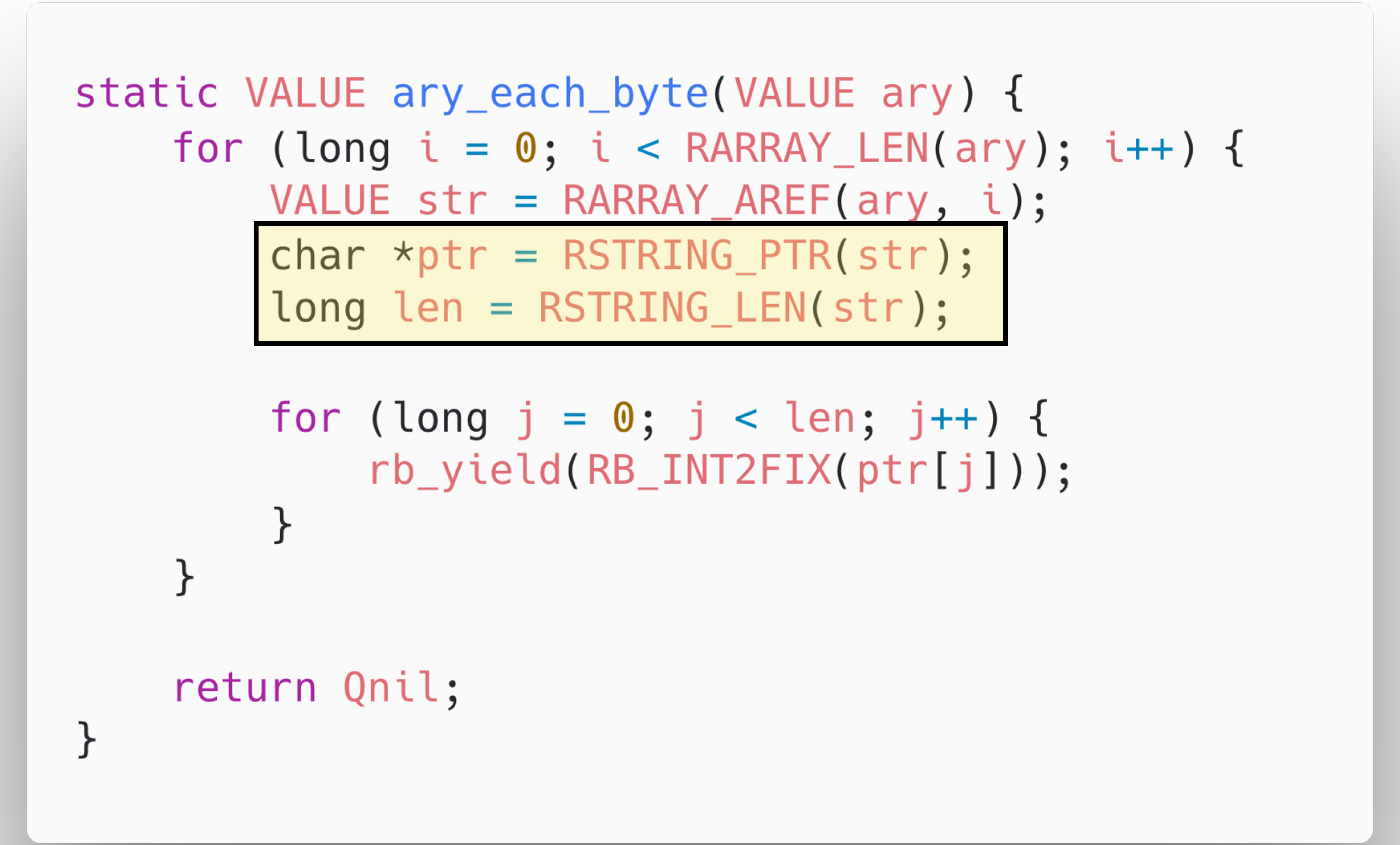
```c
static VALUE ary_each_byte(VALUE ary) {
    for (long i = 0; i < RARRAY_LEN(ary); i++) {
        VALUE str = RARRAY_AREF(ary, i);
        char *ptr = RSTRING_PTR(str);
        long len = RSTRING_LEN(str);

        for (long j = 0; j < len; j++) {
            rb_yield(RB_INT2FIX(ptr[j]));
        }
    }

    return Qnil;
}
```

Array#each_byte



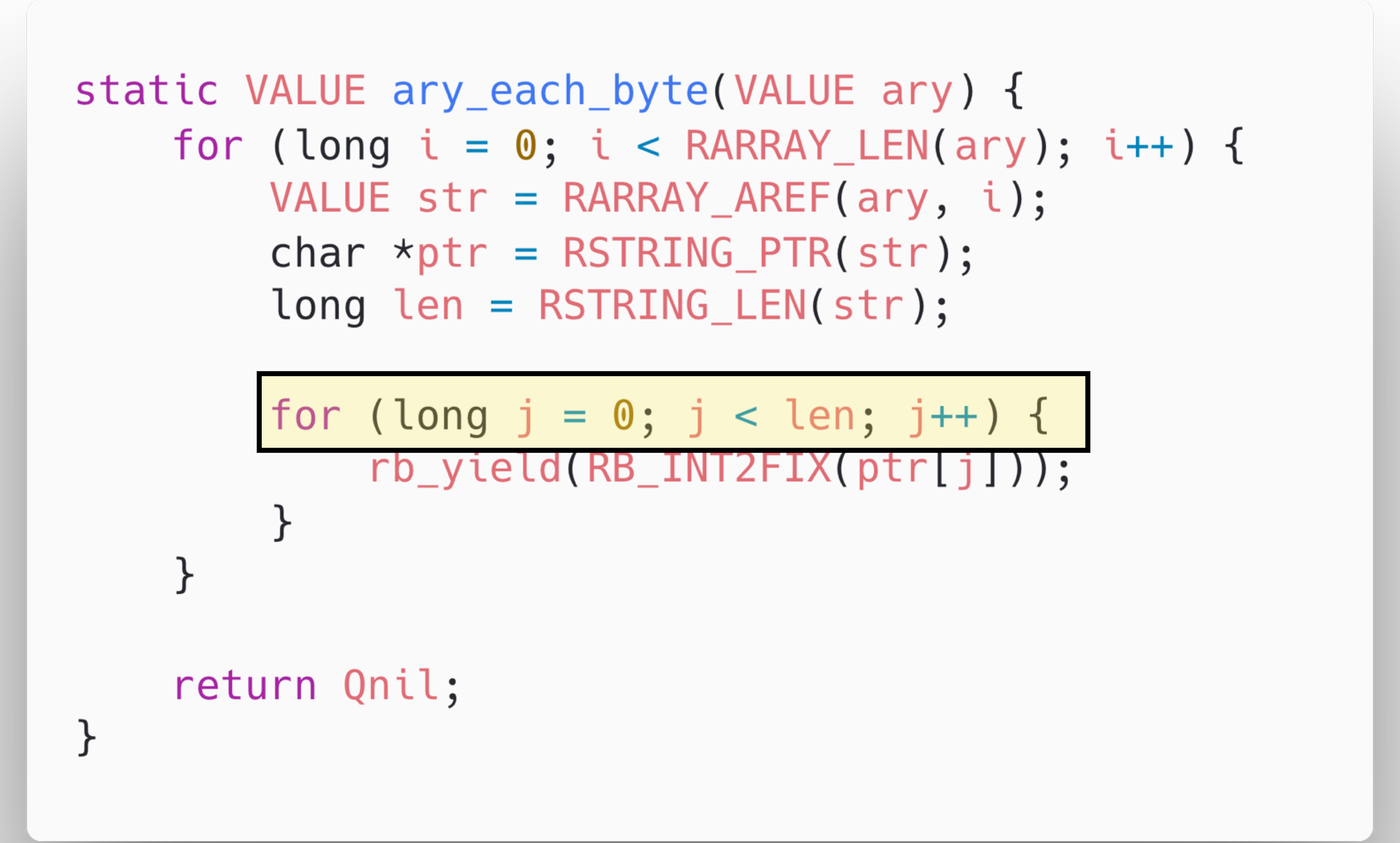
```c
static VALUE ary_each_byte(VALUE ary) {
    for (long i = 0; i < RARRAY_LEN(ary); i++) {
        VALUE str = RARRAY_AREF(ary, i);
        char *ptr = RSTRING_PTR(str);
        long len = RSTRING_LEN(str);

        for (long j = 0; j < len; j++) {
            rb_yield(RB_INT2FIX(ptr[j]));
        }
    }

    return Qnil;
}
```

Array#each_byte



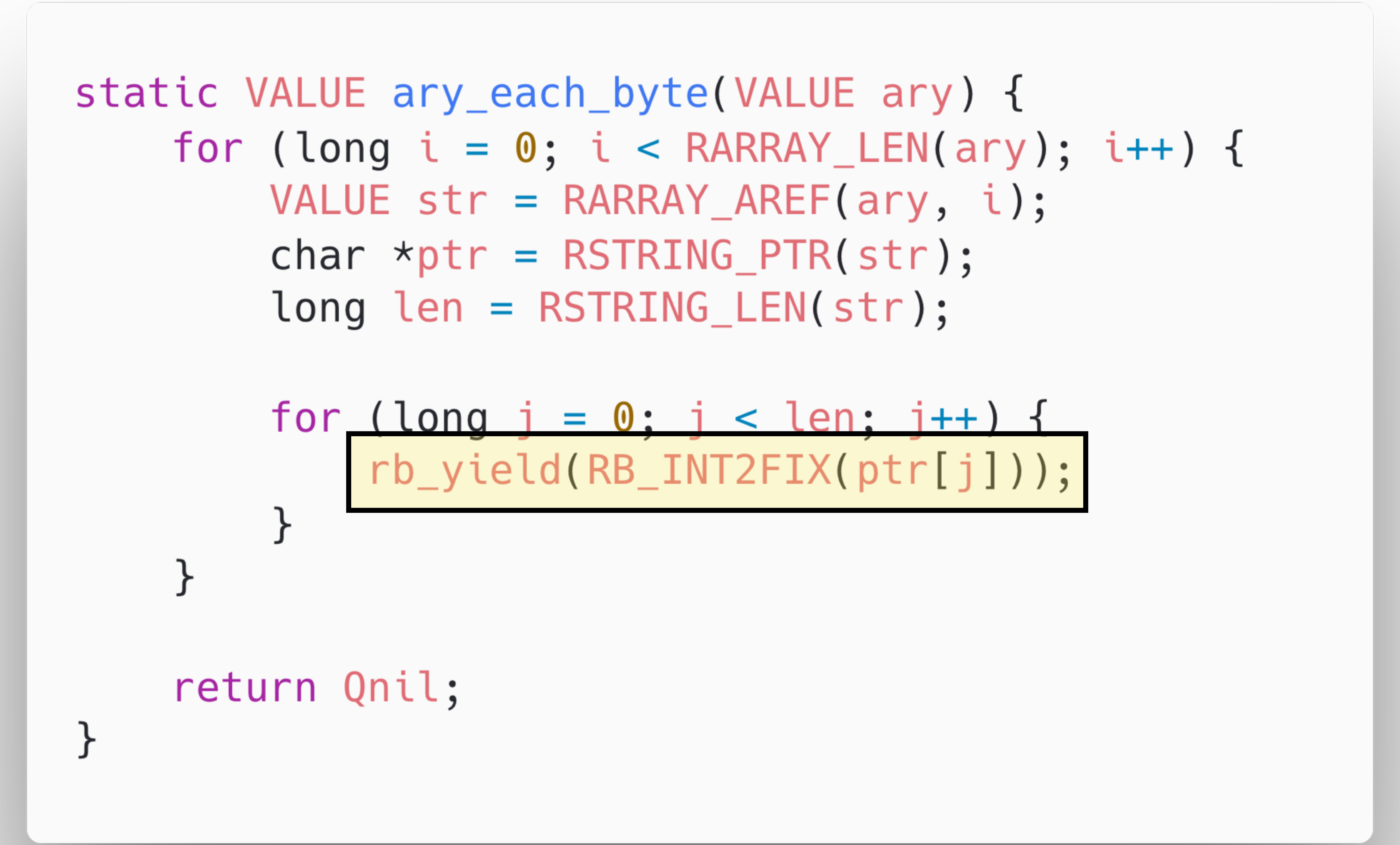
```
static VALUE ary_each_byte(VALUE ary) {
    for (long i = 0; i < RARRAY_LEN(ary); i++) {
        VALUE str = RARRAY_AREF(ary, i);
        char *ptr = RSTRING_PTR(str);
        long len = RSTRING_LEN(str);

        for (long j = 0; j < len; j++) {
            rb_yield(RB_INT2FIX(ptr[j]));
        }
    }

    return Qnil;
}
```

Array#each_byte



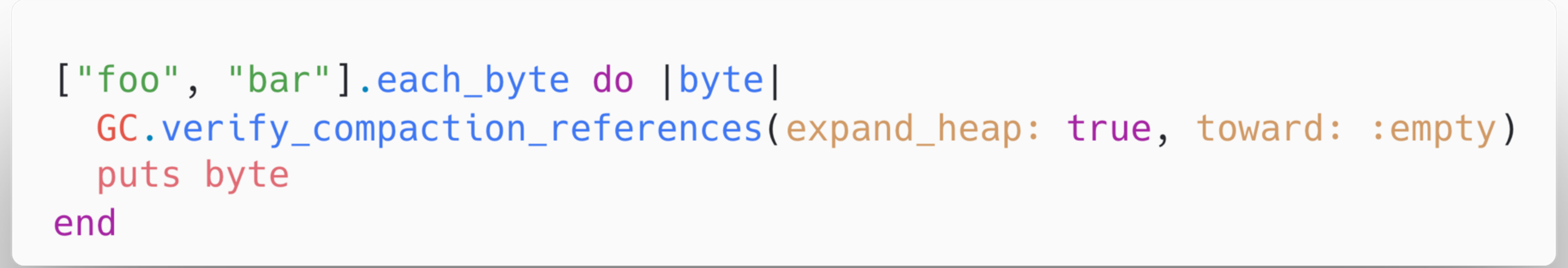
```c
static VALUE ary_each_byte(VALUE ary) {
    for (long i = 0; i < RARRAY_LEN(ary); i++) {
        VALUE str = RARRAY_AREF(ary, i);
        char *ptr = RSTRING_PTR(str);
        long len = RSTRING_LEN(str);

        for (long j = 0; j < len; j++) {
            rb_yield(RB_INT2FIX(ptr[j]));
        }
    }

    return Qnil;
}
```

Array#each_byte



```
static VALUE ary_each_byte(VALUE ary) {
    for (long i = 0; i < RARRAY_LEN(ary); i++) {
        VALUE str = RARRAY_AREF(ary, i);
        char *ptr = RSTRING_PTR(str);
        long len = RSTRING_LEN(str);

        for (long j = 0; j < len; j++) {
            rb_yield(RB_INT2FIX(ptr[j]));
        }
    }

    return Qnil;
}
```

Array#each_byte



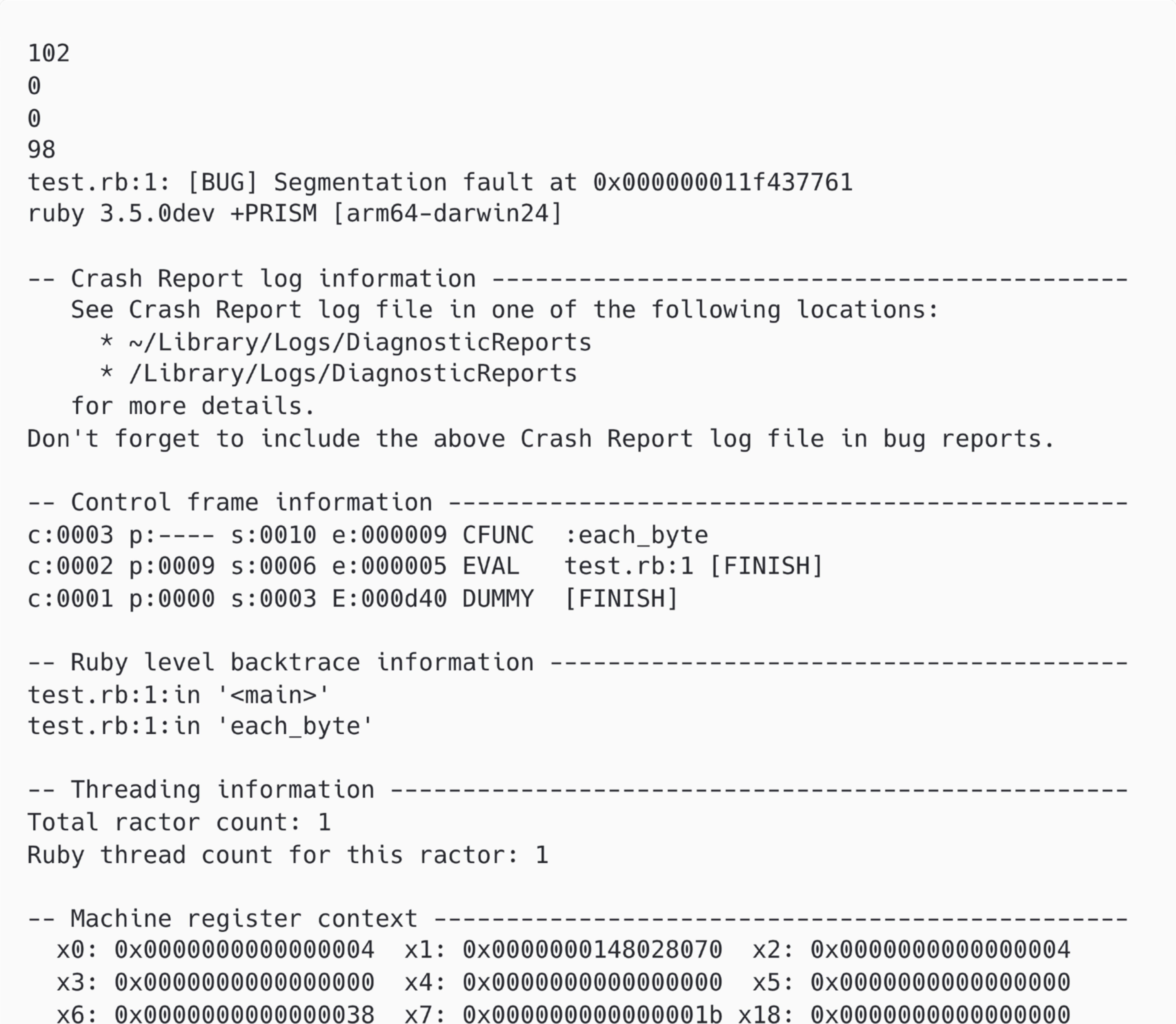
```ruby
["foo", "bar"].each_byte do |byte|
  GC.verify_compaction_references(expand_heap: true, toward: :empty)
  puts byte
end
```

Array#each_byte

```ruby
["foo", "bar"].each_byte do |byte|
  GC.verify_compaction_references(expand_heap: true, toward: :empty)
  puts byte
end
```

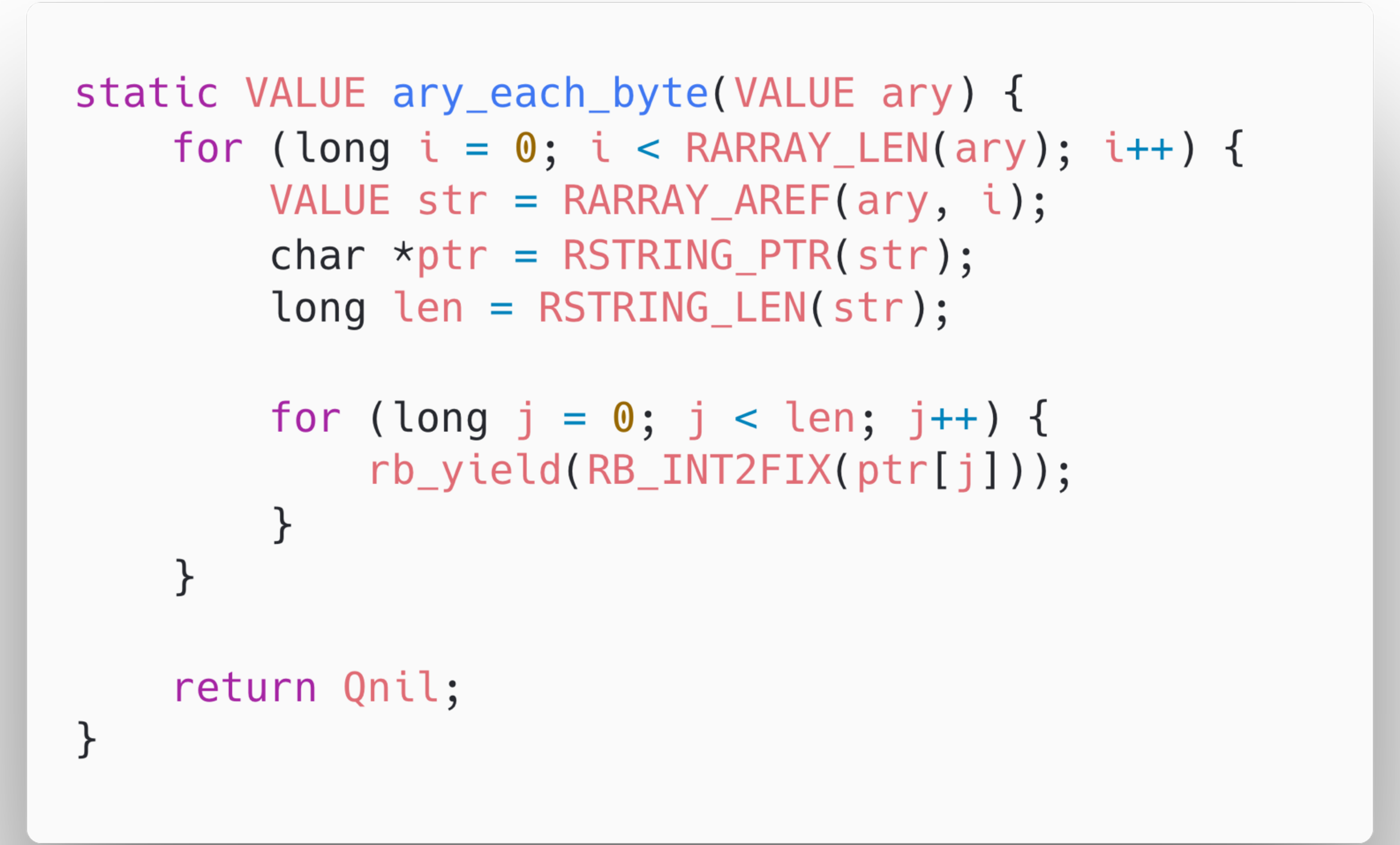
```
102
0
0
98
0
0
```

Array#each_byte



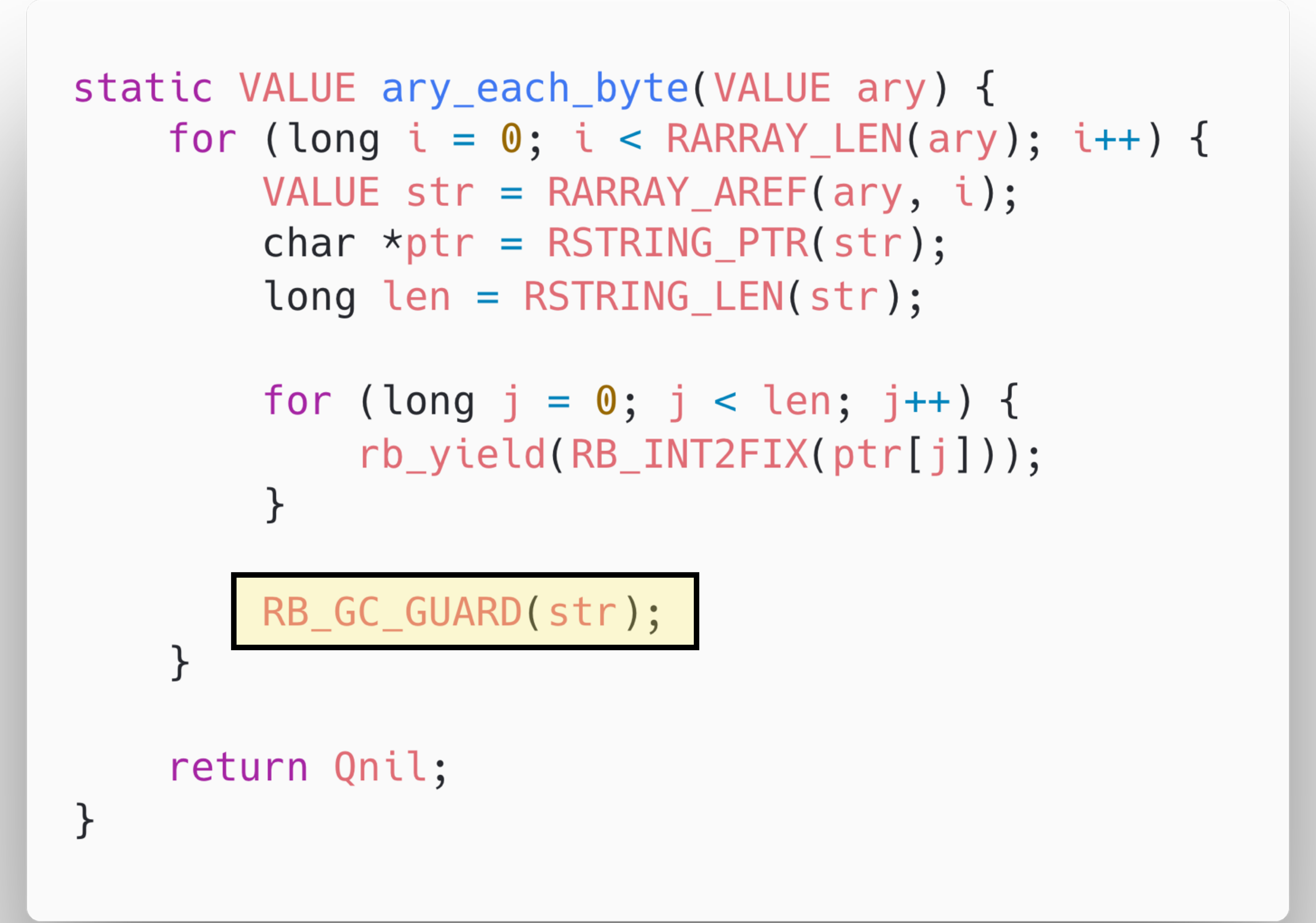
```
102
0
0
98
test.rb:1: [BUG] Segmentation fault at 0x000000011f437761
ruby 3.5.0dev +PRISM [arm64-darwin24]

-- Crash Report log information --------------------------------------------
   See Crash Report log file in one of the following locations:
     * ~/Library/Logs/DiagnosticReports
     * /Library/Logs/DiagnosticReports
   for more details.
Don't forget to include the above Crash Report log file in bug reports.

-- Control frame information -----------------------------------------------
c:0003 p:---- s:0010 e:000009 CFUNC  :each_byte
c:0002 p:0009 s:0006 e:000005 EVAL   test.rb:1 [FINISH]
c:0001 p:0000 s:0003 E:000d40 DUMMY  [FINISH]

-- Ruby level backtrace information ----------------------------------------
test.rb:1:in '<main>'
test.rb:1:in 'each_byte'

-- Threading information ---------------------------------------------------
Total ractor count: 1
Ruby thread count for this ractor: 1

-- Machine register context ------------------------------------------------
  x0: 0x0000000000000004  x1: 0x0000000148028070  x2: 0x0000000000000004
  x3: 0x0000000000000000  x4: 0x0000000000000000  x5: 0x0000000000000000
  x6: 0x0000000000000038  x7: 0x000000000000001b x18: 0x0000000000000000
```

Array#each_byte



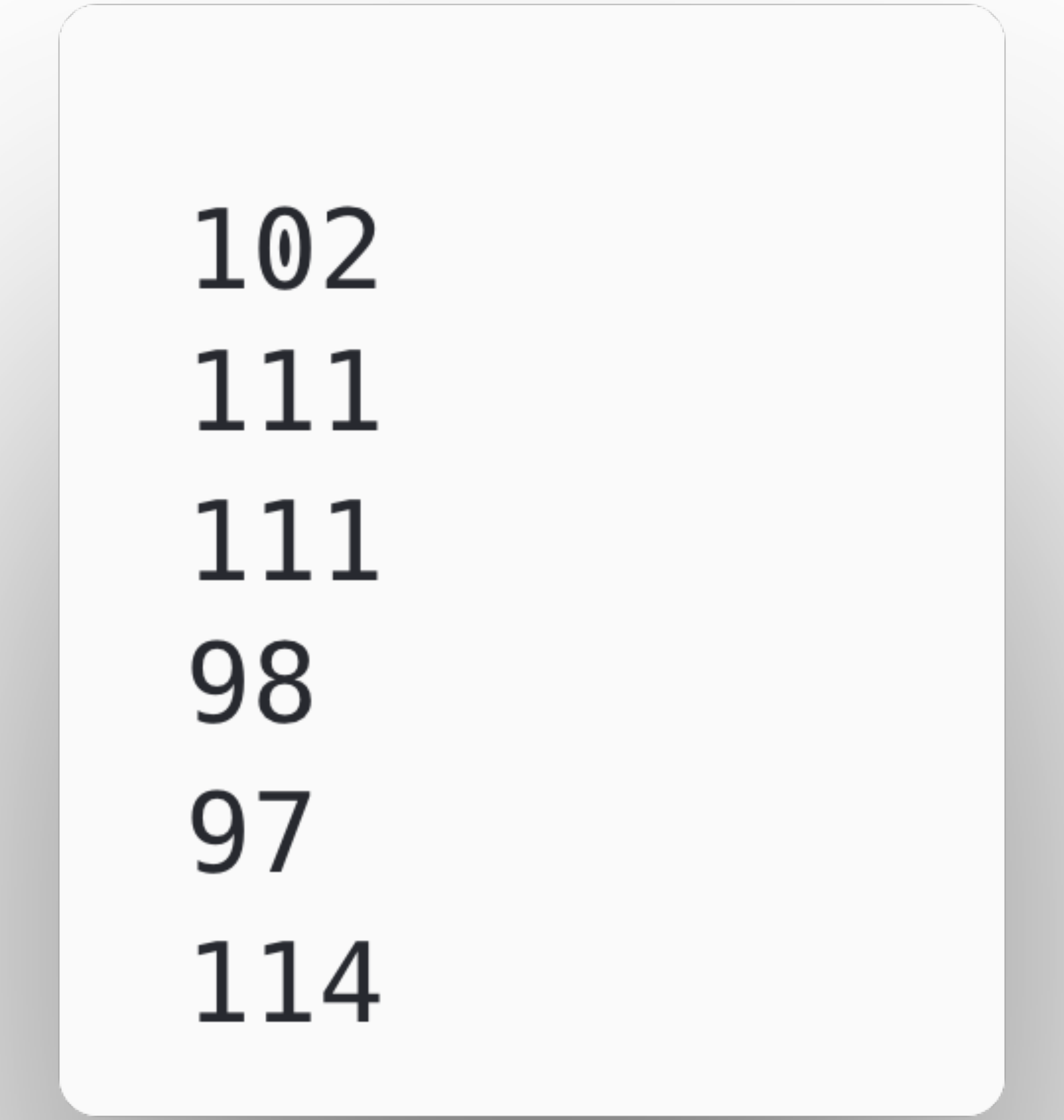
```c
static VALUE ary_each_byte(VALUE ary) {
    for (long i = 0; i < RARRAY_LEN(ary); i++) {
        VALUE str = RARRAY_AREF(ary, i);
        char *ptr = RSTRING_PTR(str);
        long len = RSTRING_LEN(str);

        for (long j = 0; j < len; j++) {
            rb_yield(RB_INT2FIX(ptr[j]));
        }
    }

    return Qnil;
}
```

Array#each_byte



```c
static VALUE ary_each_byte(VALUE ary) {
    for (long i = 0; i < RARRAY_LEN(ary); i++) {
        VALUE str = RARRAY_AREF(ary, i);
        char *ptr = RSTRING_PTR(str);
        long len = RSTRING_LEN(str);

        for (long j = 0; j < len; j++) {
            rb_yield(RB_INT2FIX(ptr[j]));
        }

        RB_GC_GUARD(str);
    }

    return Qnil;
}
```

Array#each_byte



```
102
111
111
98
97
114
```

Raising Errors Could Cause Memory Leaks

Missing Write Barriers

Rust Gems Don’t Solve These Problems

Ruby Has a C API,
Not a Rust API

Sources of Instability in Ruby & Native Gems

Bugs in C Code

- Use-After-Free
- Buffer Overflow
- Memory Leak

Incorrect Use of Ruby’s C API

- Missing GC Guard
- Raising Errors
- Missing Write Barriers

Preventing Crashes in Production

Running Debug Builds of Ruby in CI

Compiling with Assertions

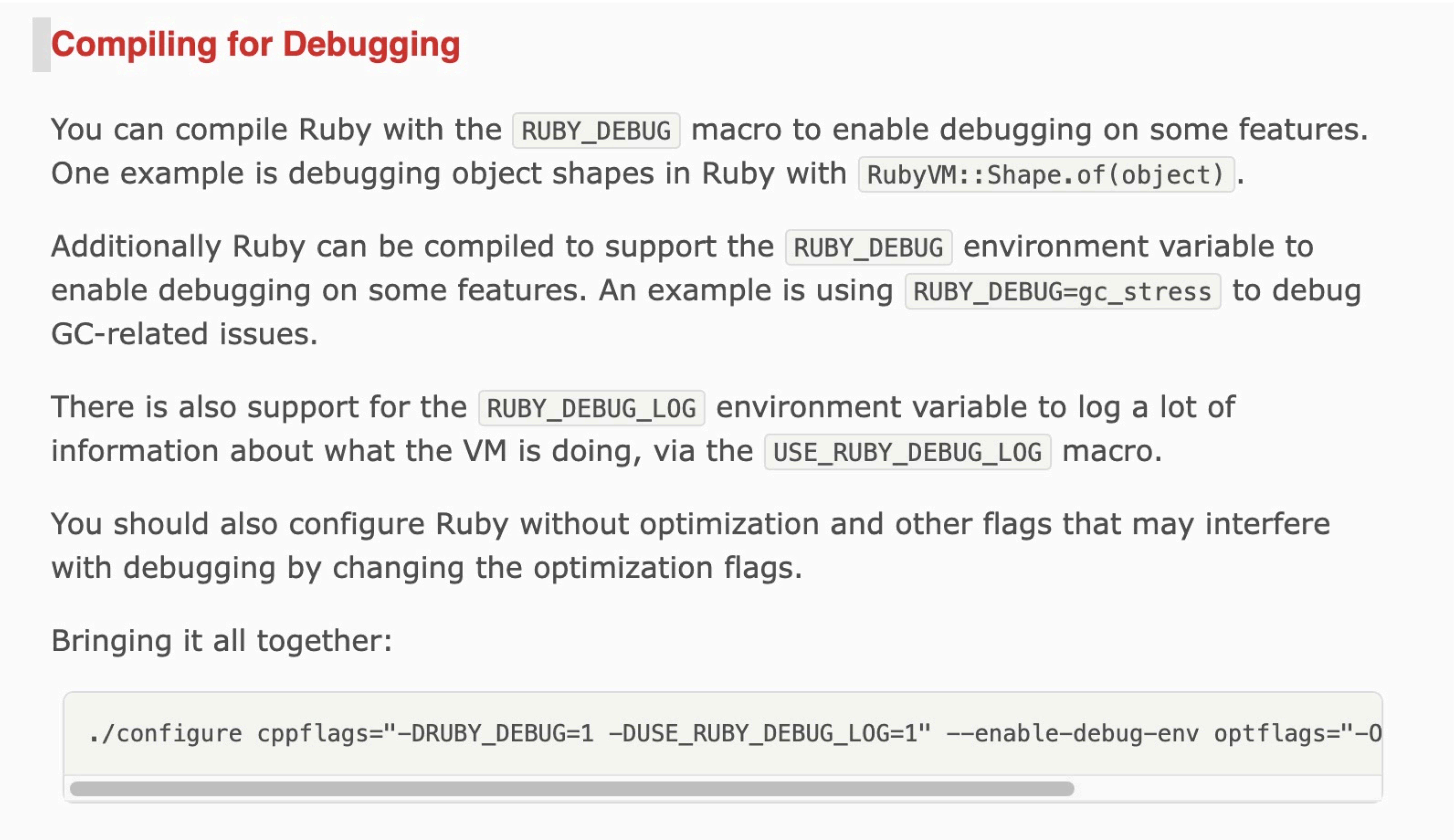
Configure Ruby with `cppflags=-DRUBY_DEBUG`

Configure Ruby with `cppflags=-DRUBY_DEBUG`

ruby-build: `CONFIGURE_OPTS='cppflags=-DRUBY_DEBUG'`

Configure Ruby with `cppflags=-DRUBY_DEBUG`

ruby-install: `ruby-install 3.4.5 -- cppflags=-DRUBY_DEBUG` docs.ruby-lang.org/en/master/contributing/building_ruby_md.html#label-Compiling+for+Debugging



Compiling for Debugging

You can compile Ruby with the `RUBY_DEBUG` macro to enable debugging on some features. One example is debugging object shapes in Ruby with `RubyVM::Shape.of(object)`.

Additionally Ruby can be compiled to support the `RUBY_DEBUG` environment variable to enable debugging on some features. An example is using `RUBY_DEBUG=gc_stress` to debug GC-related issues.

There is also support for the `RUBY_DEBUG_LOG` environment variable to log a lot of information about what the VM is doing, via the `USE_RUBY_DEBUG_LOG` macro.

You should also configure Ruby without optimization and other flags that may interfere with debugging by changing the optimization flags.

Bringing it all together:

```
./configure cppflags="-DRUBY_DEBUG=1 -DUSE_RUBY_DEBUG_LOG=1" --enable-debug-env optflags="-O
```

Running YJIT with `--yjit-call-threshold=1`

Testing With Debug Tools

Testing With Debug Tools

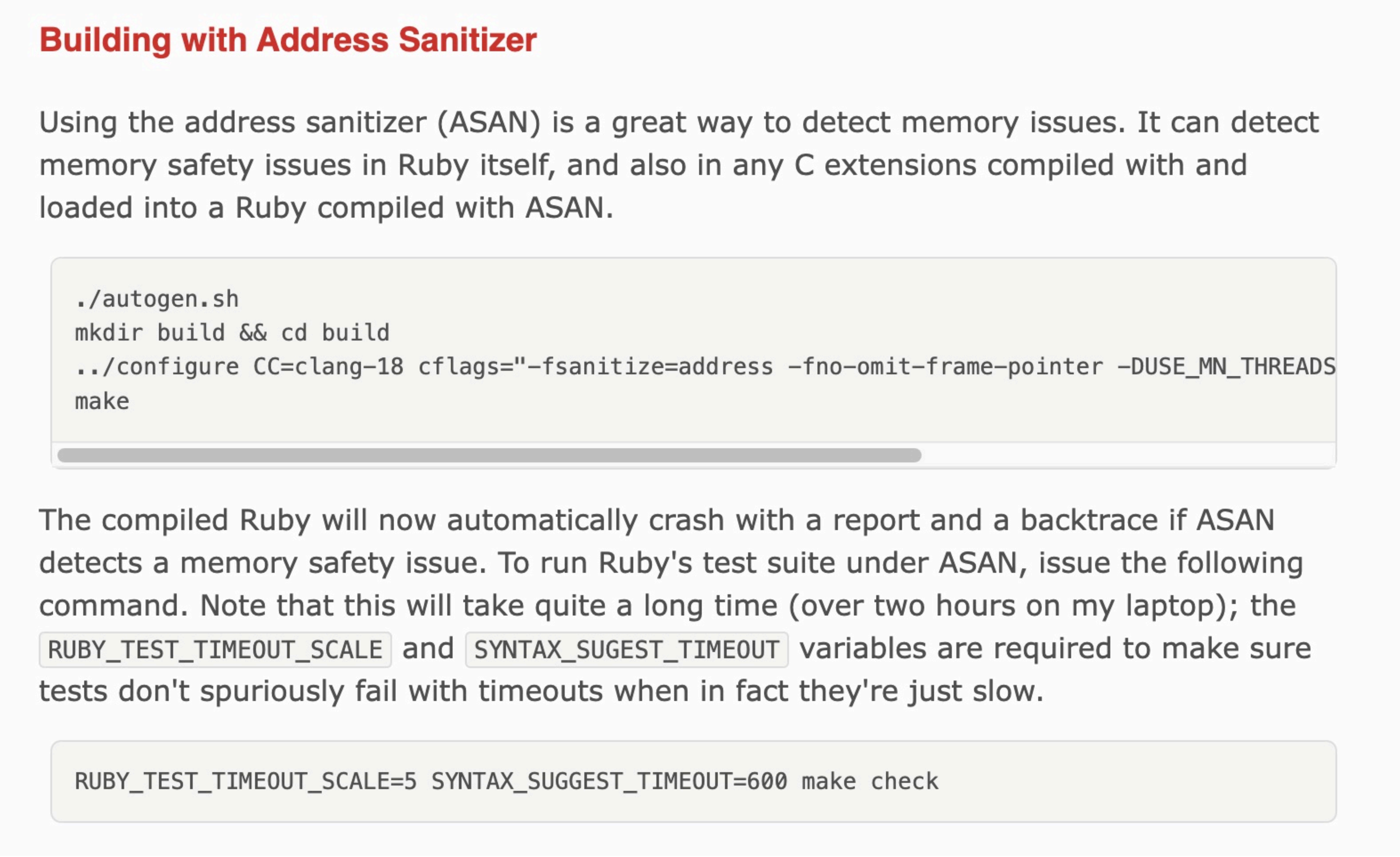
Valgrind

Address Sanitizer

```
==3911864==ERROR: AddressSanitizer: use-after-poison on address 0x7d72f67fc268 at pc 0x5c79775364bc bp 0x7ffd105a1910 sp
0x7ffd105a1908
READ of size 8 at 0x7d72f67fc268 thread T0
    #0 0x5c79775364bb in RB_BUILTIN_TYPE /tmp/ruby/src/trunk_asan/include/ruby/internal/value_type.h:191:30
    #1 0x5c79775364bb in rb_gc_shutdown_call_finalizer_p /tmp/ruby/src/trunk_asan/gc.c:357:18
    #2 0x5c79775364bb in rb_gc_impl_shutdown_call_finalizer /tmp/ruby/src/trunk_asan/gc/default/default.c:3045:21
    #3 0x5c79775364bb in rb_objspace_call_finalizer /tmp/ruby/src/trunk_asan/gc.c:1739:5
    #4 0x5c79775051b2 in rb_ec_finalize /tmp/ruby/src/trunk_asan/eval.c:165:5
    #5 0x5c79775051b2 in rb_ec_cleanup /tmp/ruby/src/trunk_asan/eval.c:256:5
    #6 0x5c7977505d68 in ruby_run_node /tmp/ruby/src/trunk_asan/eval.c:320:12
    #7 0x5c79774fec36 in rb_main /tmp/ruby/src/trunk_asan/main.c:42:12
    #8 0x5c79774fec36 in main /tmp/ruby/src/trunk_asan/main.c:62:12
    #9 0x7d72f702a1c9 in __libc_start_call_main csu/../sysdeps/nptl/libc_start_call_main.h:58:16
    #10 0x7d72f702a28a in __libc_start_main csu/../csu/libc-start.c:360:3
    #11 0x5c7977422a04 in _start (/tmp/ruby/build/trunk_asan/ruby+0x16ba04) (BuildId:
6912b36e90dff44bc034130a379a6c13f4cc9936)
Address 0x7d72f67fc268 is a wild pointer inside of access range of size 0x000000000008.
```

Testing With Debug Tools

docs.ruby-lang.org/en/master/contributing/building_ruby_md.html#label-Building+with+Address+Sanitizer



Building with Address Sanitizer

Using the address sanitizer (ASAN) is a great way to detect memory issues. It can detect memory safety issues in Ruby itself, and also in any C extensions compiled with and loaded into a Ruby compiled with ASAN.

```
./autogen.sh
mkdir build && cd build
../configure CC=clang-18 cflags="-fsanitize=address -fno-omit-frame-pointer -DUSE_MN_THREADS
make
```

The compiled Ruby will now automatically crash with a report and a backtrace if ASAN detects a memory safety issue. To run Ruby's test suite under ASAN, issue the following command. Note that this will take quite a long time (over two hours on my laptop); the `RUBY_TEST_TIMEOUT_SCALE` and `SYNTAX_SUGEST_TIMEOUT` variables are required to make sure tests don't spuriously fail with timeouts when in fact they're just slow.

```
RUBY_TEST_TIMEOUT_SCALE=5 SYNTAX_SUGGEST_TIMEOUT=600 make check
```

Testing on Ruby-Head

Running Various Configurations on Nightly CI

Preventing Crashes in Production

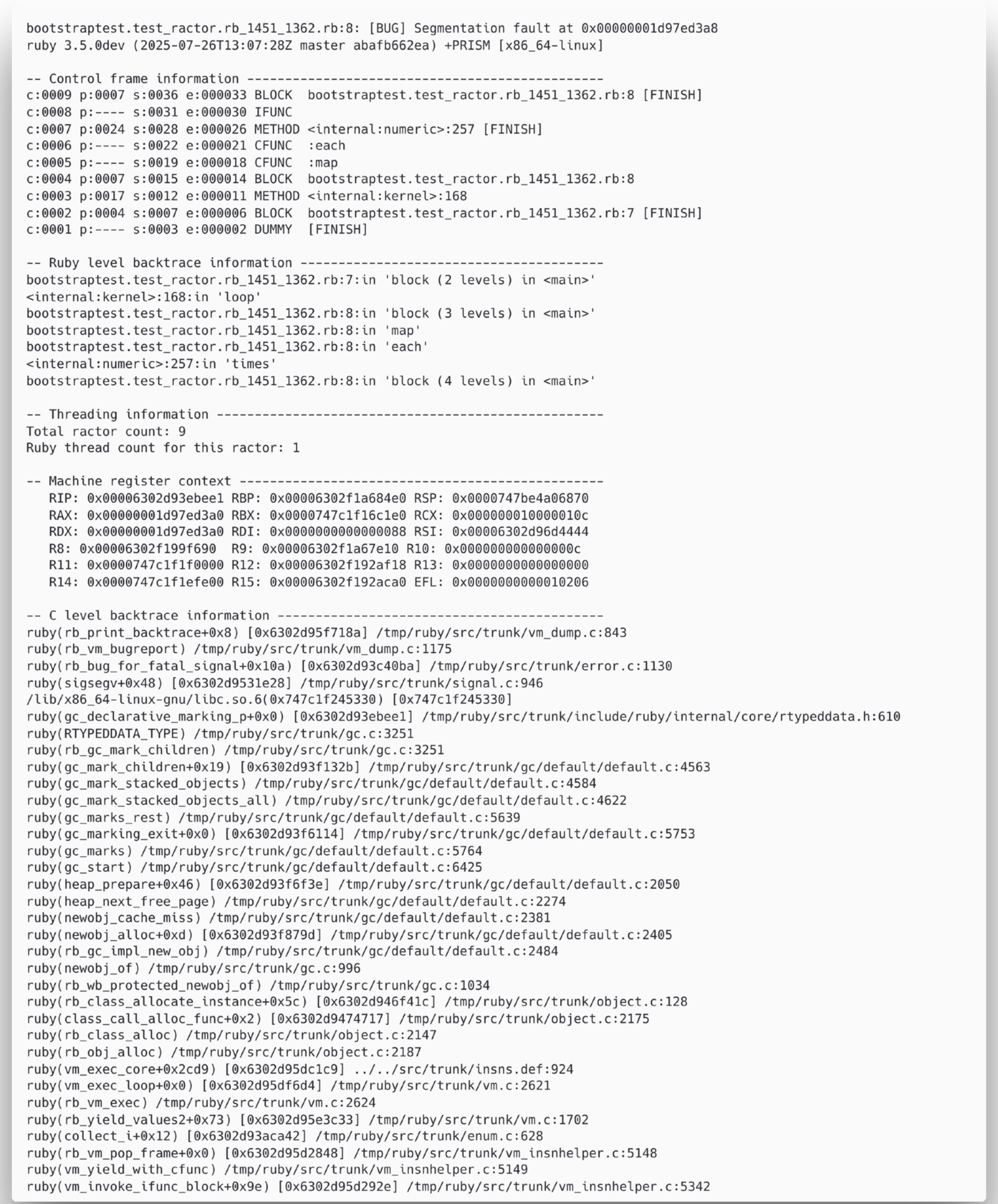
Compiling with Assertions

Running with YJIT Call Threshold of 1

Testing with Valgrind/ASAN

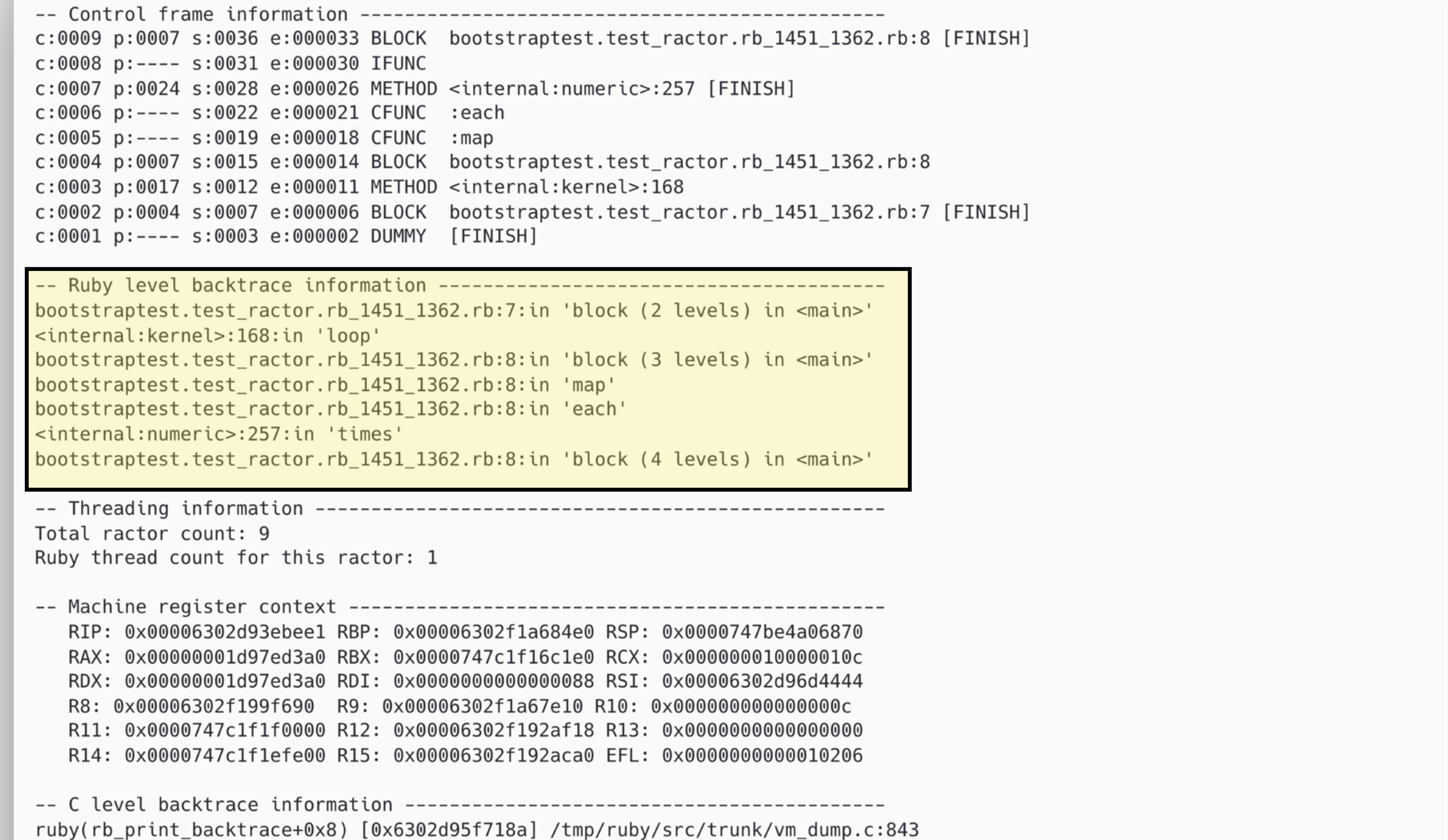
Capturing Information About Crashes

Collecting Crash Reports



```
bootstraptest.test_ractor.rb_1451_1362.rb:8: [BUG] Segmentation fault at 0x00000001d97ed3a8
ruby 3.5.0dev (2025-07-26T13:07:28Z master abafb662ea) +PRISM [x86_64-linux]

-- Control frame information -----------------------------------------------
c:0009 p:0007 s:0036 e:000033 BLOCK  bootstraptest.test_ractor.rb_1451_1362.rb:8 [FINISH]
c:0008 p:---- s:0031 e:000030 IFUNC
c:0007 p:0024 s:0028 e:000026 METHOD <internal:numeric>:257 [FINISH]
c:0006 p:---- s:0022 e:000021 CFUNC  :each
c:0005 p:---- s:0019 e:000018 CFUNC  :map
c:0004 p:0007 s:0015 e:000014 BLOCK  bootstraptest.test_ractor.rb_1451_1362.rb:8
c:0003 p:0017 s:0012 e:000011 METHOD <internal:kernel>:168
c:0002 p:0004 s:0007 e:000006 BLOCK  bootstraptest.test_ractor.rb_1451_1362.rb:7 [FINISH]
c:0001 p:---- s:0003 e:000002 DUMMY  [FINISH]

-- Ruby level backtrace information ----------------------------------------
bootstraptest.test_ractor.rb_1451_1362.rb:7:in 'block (2 levels) in <main>'
<internal:kernel>:168:in 'loop'
bootstraptest.test_ractor.rb_1451_1362.rb:8:in 'block (3 levels) in <main>'
bootstraptest.test_ractor.rb_1451_1362.rb:8:in 'map'
bootstraptest.test_ractor.rb_1451_1362.rb:8:in 'each'
<internal:numeric>:257:in 'times'
bootstraptest.test_ractor.rb_1451_1362.rb:8:in 'block (4 levels) in <main>'

-- Threading information ---------------------------------------------------
Total ractor count: 9
Ruby thread count for this ractor: 1

-- Machine register context ------------------------------------------------
   RIP: 0x00006302d93ebee1 RBP: 0x00006302f1a684e0 RSP: 0x0000747be4a06870
   RAX: 0x00000001d97ed3a0 RBX: 0x0000747c1f16c1e0 RCX: 0x000000010000010c
   RDX: 0x00000001d97ed3a0 RDI: 0x0000000000000088 RSI: 0x00006302d96d4444
   R8: 0x00006302f199f690  R9: 0x00006302f1a67e10 R10: 0x000000000000000c
   R11: 0x0000747c1f1f0000 R12: 0x00006302f192af18 R13: 0x0000000000000000
   R14: 0x0000747c1f1efe00 R15: 0x00006302f192aca0 EFL: 0x0000000000010206

-- C level backtrace information -------------------------------------------
ruby(rb_print_backtrace+0x8) [0x6302d95f718a] /tmp/ruby/src/trunk/vm_dump.c:843
ruby(rb_vm_bugreport) /tmp/ruby/src/trunk/vm_dump.c:1175
ruby(rb_bug_for_fatal_signal+0x10a) [0x6302d93c40ba] /tmp/ruby/src/trunk/error.c:1130
ruby(sigsegv+0x48) [0x6302d9531e28] /tmp/ruby/src/trunk/signal.c:946
/lib/x86_64-linux-gnu/libc.so.6(0x747c1f245330) [0x747c1f245330]
ruby(gc_declarative_marking_p+0x0) [0x6302d93ebee1] /tmp/ruby/src/trunk/include/ruby/internal/core/rtypeddata.h:610
ruby(RTYPEDDATA_TYPE) /tmp/ruby/src/trunk/gc.c:3251
ruby(rb_gc_mark_children) /tmp/ruby/src/trunk/gc.c:3251
ruby(gc_mark_children+0x19) [0x6302d93f132b] /tmp/ruby/src/trunk/gc/default/default.c:4563
ruby(gc_mark_stacked_objects) /tmp/ruby/src/trunk/gc/default/default.c:4584
ruby(gc_mark_stacked_objects_all) /tmp/ruby/src/trunk/gc/default/default.c:4622
ruby(gc_marks_rest) /tmp/ruby/src/trunk/gc/default/default.c:5639
ruby(gc_marking_exit+0x0) [0x6302d93f6114] /tmp/ruby/src/trunk/gc/default/default.c:5753
ruby(gc_marks) /tmp/ruby/src/trunk/gc/default/default.c:5764
ruby(gc_start) /tmp/ruby/src/trunk/gc/default/default.c:6425
ruby(heap_prepare+0x46) [0x6302d93f6f3e] /tmp/ruby/src/trunk/gc/default/default.c:2050
ruby(heap_next_free_page) /tmp/ruby/src/trunk/gc/default/default.c:2274
ruby(newobj_cache_miss) /tmp/ruby/src/trunk/gc/default/default.c:2381
ruby(newobj_alloc+0xd) [0x6302d93f879d] /tmp/ruby/src/trunk/gc/default/default.c:2405
ruby(rb_gc_impl_new_obj) /tmp/ruby/src/trunk/gc/default/default.c:2484
ruby(newobj_of) /tmp/ruby/src/trunk/gc.c:996
ruby(rb_wb_protected_newobj_of) /tmp/ruby/src/trunk/gc.c:1034
ruby(rb_class_allocate_instance+0x5c) [0x6302d946f41c] /tmp/ruby/src/trunk/object.c:128
ruby(class_call_alloc_func+0x2) [0x6302d9474717] /tmp/ruby/src/trunk/object.c:2175
ruby(rb_class_alloc) /tmp/ruby/src/trunk/object.c:2147
ruby(rb_obj_alloc) /tmp/ruby/src/trunk/object.c:2187
ruby(vm_exec_core+0x2cd9) [0x6302d95dc1c9] ../../src/trunk/insns.def:924
ruby(vm_exec_loop+0x0) [0x6302d95df6d4] /tmp/ruby/src/trunk/vm.c:2621
ruby(rb_vm_exec) /tmp/ruby/src/trunk/vm.c:2624
ruby(rb_yield_values2+0x73) [0x6302d95e3c33] /tmp/ruby/src/trunk/vm.c:1702
ruby(collect_i+0x12) [0x6302d93aca42] /tmp/ruby/src/trunk/enum.c:628
ruby(rb_vm_pop_frame+0x0) [0x6302d95d2848] /tmp/ruby/src/trunk/vm_insnhelper.c:5148
ruby(vm_yield_with_cfunc) /tmp/ruby/src/trunk/vm_insnhelper.c:5149
ruby(vm_invoke_ifunc_block+0x9e) [0x6302d95d292e] /tmp/ruby/src/trunk/vm_insnhelper.c:5342
```

```
bootstraptest.test_ractor.rb_1451_1362.rb:8: [BUG] Segmentation fault at 0x00000001d97ed3a8
ruby 3.5.0dev (2025-07-26T13:07:28Z master abafb662ea) +PRISM [x86_64-linux]

-- Control frame information -----------------------------------------------
c:0009 p:0007 s:0036 e:000033 BLOCK  bootstraptest.test_ractor.rb_1451_1362.rb:8 [FINISH]
c:0008 p:---- s:0031 e:000030 IFUNC
c:0007 p:0024 s:0028 e:000026 METHOD <internal:numeric>:257 [FINISH]
c:0006 p:---- s:0022 e:000021 CFUNC  :each
c:0005 p:---- s:0019 e:000018 CFUNC  :map
c:0004 p:0007 s:0015 e:000014 BLOCK  bootstraptest.test_ractor.rb_1451_1362.rb:8
c:0003 p:0017 s:0012 e:000011 METHOD <internal:kernel>:168
c:0002 p:0004 s:0007 e:000006 BLOCK  bootstraptest.test_ractor.rb_1451_1362.rb:7 [FINISH]
c:0001 p:---- s:0003 e:000002 DUMMY  [FINISH]

-- Ruby level backtrace information ----------------------------------------
bootstraptest.test_ractor.rb_1451_1362.rb:7:in 'block (2 levels) in <main>'
<internal:kernel>:168:in 'loop'
bootstraptest.test_ractor.rb_1451_1362.rb:8:in 'block (3 levels) in <main>'
bootstraptest.test_ractor.rb_1451_1362.rb:8:in 'map'
bootstraptest.test_ractor.rb_1451_1362.rb:8:in 'each'
<internal:numeric>:257:in 'times'
bootstraptest.test_ractor.rb_1451_1362.rb:8:in 'block (4 levels) in <main>'
```

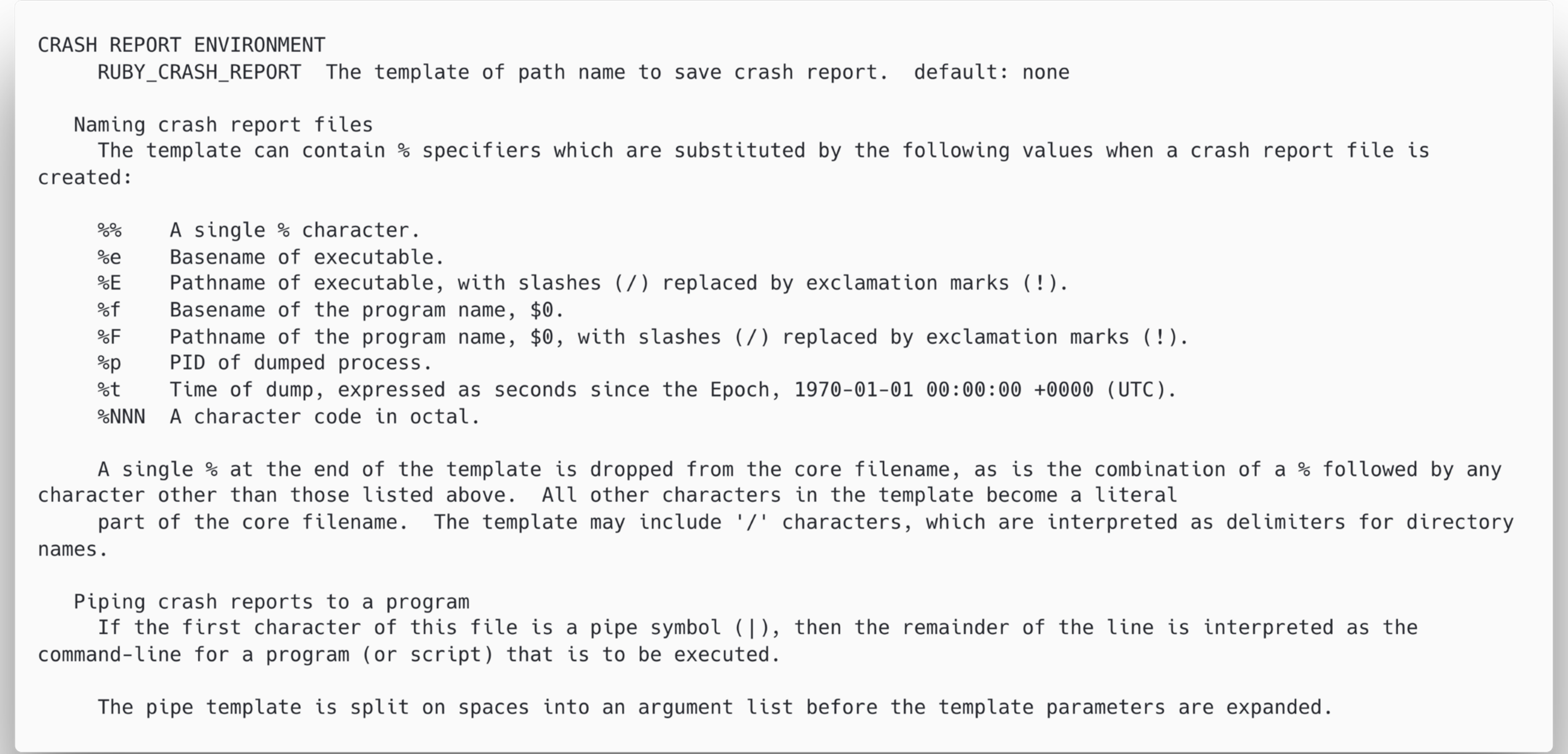


```
-- Control frame information -----------------------------------------------
c:0009 p:0007 s:0036 e:000033 BLOCK  bootstraptest.test_ractor.rb_1451_1362.rb:8 [FINISH]
c:0008 p:---- s:0031 e:000030 IFUNC
c:0007 p:0024 s:0028 e:000026 METHOD <internal:numeric>:257 [FINISH]
c:0006 p:---- s:0022 e:000021 CFUNC  :each
c:0005 p:---- s:0019 e:000018 CFUNC  :map
c:0004 p:0007 s:0015 e:000014 BLOCK  bootstraptest.test_ractor.rb_1451_1362.rb:8
c:0003 p:0017 s:0012 e:000011 METHOD <internal:kernel>:168
c:0002 p:0004 s:0007 e:000006 BLOCK  bootstraptest.test_ractor.rb_1451_1362.rb:7 [FINISH]
c:0001 p:---- s:0003 e:000002 DUMMY  [FINISH]

-- Ruby level backtrace information ----------------------------------------
bootstraptest.test_ractor.rb_1451_1362.rb:7:in 'block (2 levels) in <main>'
<internal:kernel>:168:in 'loop'
bootstraptest.test_ractor.rb_1451_1362.rb:8:in 'block (3 levels) in <main>'
bootstraptest.test_ractor.rb_1451_1362.rb:8:in 'map'
bootstraptest.test_ractor.rb_1451_1362.rb:8:in 'each'
<internal:numeric>:257:in 'times'
bootstraptest.test_ractor.rb_1451_1362.rb:8:in 'block (4 levels) in <main>'

-- Threading information ---------------------------------------------------
Total ractor count: 9
Ruby thread count for this ractor: 1

-- Machine register context ------------------------------------------------
   RIP: 0x00006302d93ebee1 RBP: 0x00006302f1a684e0 RSP: 0x0000747be4a06870
   RAX: 0x00000001d97ed3a0 RBX: 0x0000747c1f16c1e0 RCX: 0x000000010000010c
   RDX: 0x00000001d97ed3a0 RDI: 0x0000000000000088 RSI: 0x00006302d96d4444
   R8: 0x00006302f199f690  R9: 0x00006302f1a67e10 R10: 0x000000000000000c
   R11: 0x0000747c1f1f0000 R12: 0x00006302f192af18 R13: 0x0000000000000000
   R14: 0x0000747c1f1efe00 R15: 0x00006302f192aca0 EFL: 0x0000000000010206

-- C level backtrace information -------------------------------------------
ruby(rb_print_backtrace+0x8) [0x6302d95f718a] /tmp/ruby/src/trunk/vm_dump.c:843
```

```
  R8: 0x00006302f199f690  R9: 0x00006302f1a67e10 R10: 0x000000000000000c
 R11: 0x0000747c1f1f0000 R12: 0x00006302f192af18 R13: 0x0000000000000000
 R14: 0x0000747c1f1efe00 R15: 0x00006302f192aca0 EFL: 0x0000000000010206
```

```
-- C level backtrace information -------------------------------------------
ruby(rb_print_backtrace+0x8) [0x6302d95f718a] /tmp/ruby/src/trunk/vm_dump.c:843
ruby(rb_vm_bugreport) /tmp/ruby/src/trunk/vm_dump.c:1175
ruby(rb_bug_for_fatal_signal+0x10a) [0x6302d93c40ba] /tmp/ruby/src/trunk/error.c:1130
ruby(sigsegv+0x48) [0x6302d9531e28] /tmp/ruby/src/trunk/signal.c:946
/lib/x86_64-linux-gnu/libc.so.6(0x747c1f245330) [0x747c1f245330]
ruby(gc_declarative_marking_p+0x0) [0x6302d93ebee1] /tmp/ruby/src/trunk/include/ruby/internal/core/rtypeddata.h:610
ruby(RTYPEDDATA_TYPE) /tmp/ruby/src/trunk/gc.c:3251
ruby(rb_gc_mark_children) /tmp/ruby/src/trunk/gc.c:3251
ruby(gc_mark_children+0x19) [0x6302d93f132b] /tmp/ruby/src/trunk/gc/default/default.c:4563
ruby(gc_mark_stacked_objects) /tmp/ruby/src/trunk/gc/default/default.c:4584
ruby(gc_mark_stacked_objects_all) /tmp/ruby/src/trunk/gc/default/default.c:4622
ruby(gc_marks_rest) /tmp/ruby/src/trunk/gc/default/default.c:5639
ruby(gc_marking_exit+0x0) [0x6302d93f6114] /tmp/ruby/src/trunk/gc/default/default.c:5753
ruby(gc_marks) /tmp/ruby/src/trunk/gc/default/default.c:5764
ruby(gc_start) /tmp/ruby/src/trunk/gc/default/default.c:6425
ruby(heap_prepare+0x46) [0x6302d93f6f3e] /tmp/ruby/src/trunk/gc/default/default.c:2050
ruby(heap_next_free_page) /tmp/ruby/src/trunk/gc/default/default.c:2274
ruby(newobj_cache_miss) /tmp/ruby/src/trunk/gc/default/default.c:2381
ruby(newobj_alloc+0xd) [0x6302d93f879d] /tmp/ruby/src/trunk/gc/default/default.c:2405
ruby(rb_gc_impl_new_obj) /tmp/ruby/src/trunk/gc/default/default.c:2484
ruby(newobj_of) /tmp/ruby/src/trunk/gc.c:996
ruby(rb_wb_protected_newobj_of) /tmp/ruby/src/trunk/gc.c:1034
ruby(rb_class_allocate_instance+0x5c) [0x6302d946f41c] /tmp/ruby/src/trunk/object.c:128
ruby(class_call_alloc_func+0x2) [0x6302d9474717] /tmp/ruby/src/trunk/object.c:2175
ruby(rb_class_alloc) /tmp/ruby/src/trunk/object.c:2147
ruby(rb_obj_alloc) /tmp/ruby/src/trunk/object.c:2187
ruby(vm_exec_core+0x2cd9) [0x6302d95dc1c9] ../../src/trunk/insns.def:924
ruby(vm_exec_loop+0x0) [0x6302d95df6d4] /tmp/ruby/src/trunk/vm.c:2621
ruby(rb_vm_exec) /tmp/ruby/src/trunk/vm.c:2624
```

Redirecting Crash Reports



```
CRASH REPORT ENVIRONMENT
     RUBY_CRASH_REPORT  The template of path name to save crash report.  default: none

   Naming crash report files
     The template can contain % specifiers which are substituted by the following values when a crash report file is
created:

     %%    A single % character.
     %e    Basename of executable.
     %E    Pathname of executable, with slashes (/) replaced by exclamation marks (!).
     %f    Basename of the program name, $0.
     %F    Pathname of the program name, $0, with slashes (/) replaced by exclamation marks (!).
     %p    PID of dumped process.
     %t    Time of dump, expressed as seconds since the Epoch, 1970-01-01 00:00:00 +0000 (UTC).
     %NNN  A character code in octal.

     A single % at the end of the template is dropped from the core filename, as is the combination of a % followed by any
character other than those listed above.  All other characters in the template become a literal
     part of the core filename.  The template may include '/' characters, which are interpreted as delimiters for directory
names.

   Piping crash reports to a program
     If the first character of this file is a pipe symbol (|), then the remainder of the line is interpreted as the
command-line for a program (or script) that is to be executed.

     The pipe template is split on spaces into an argument list before the template parameters are expanded.
```

Generating Core Dumps

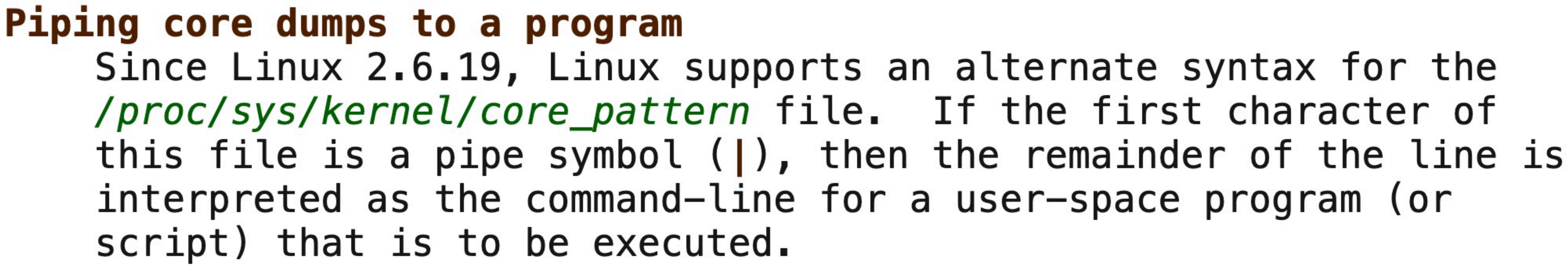
Core Dumps Contain Program State at the Time of Crash

⚠️ Core Dumps Contain Secrets ⚠️

Crash Reporter

1️⃣ Capturing Core Dumps

man7.org/linux/man-pages/man5/core.5.html



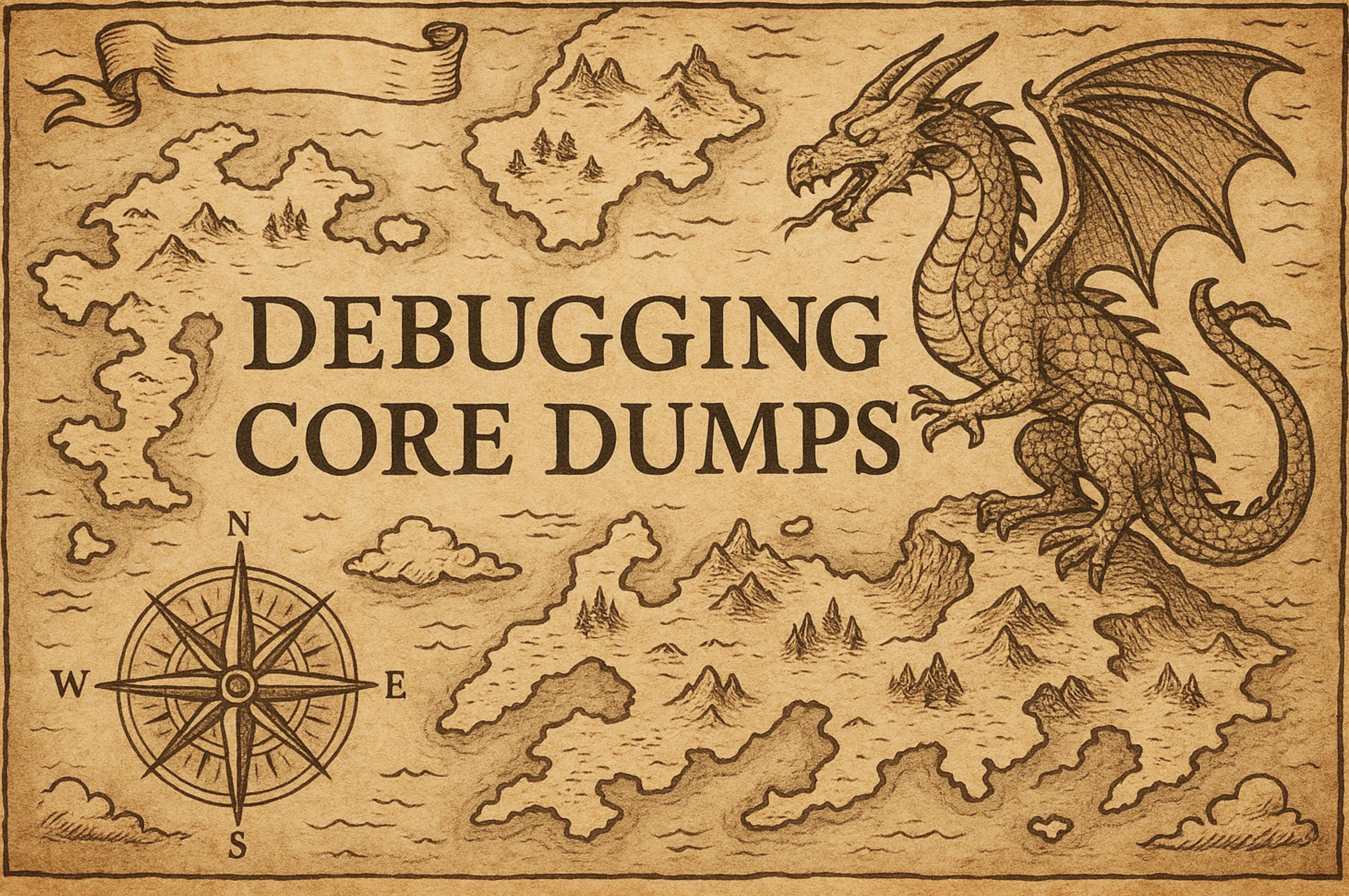
Piping core dumps to a program

Since Linux 2.6.19, Linux supports an alternate syntax for the *proc/sys/kernel/core_pattern* file. If the first character of this file is a pipe symbol (**|**), then the remainder of the line is interpreted as the command-line for a user-space program (or script) that is to be executed.

2️⃣ Capturing Crash Report

3️⃣ Error Monitoring

Debugging Core Dumps



DEBUGGING CORE DUMPS

Debugging Core Dumps

Debugging Core Dumps

Core Dump

Debugging Core Dumps

Core Dump

Original Binaries

Debugging Core Dumps

Core Dump

Original Binaries

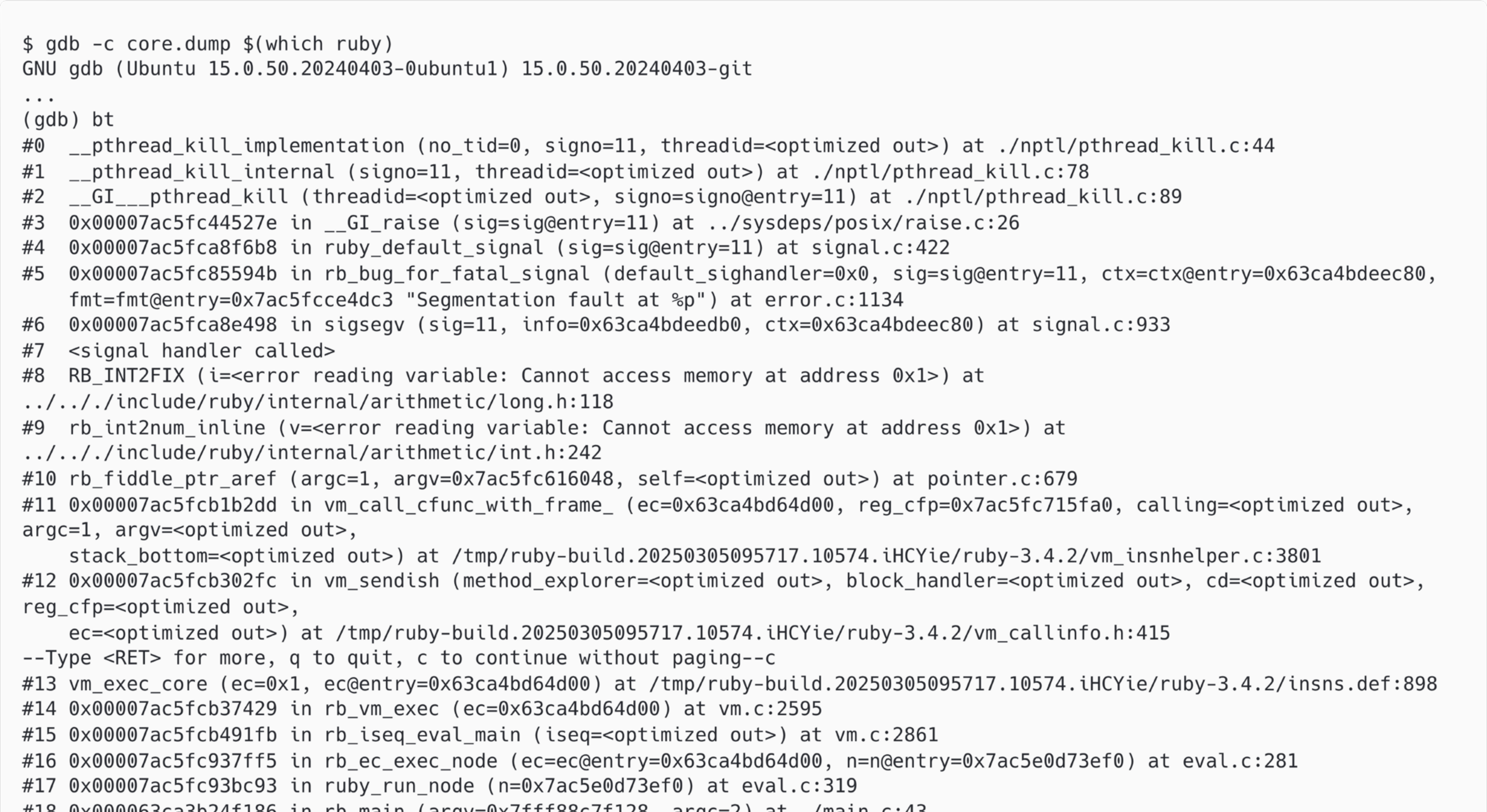
Same System

Debugging Core Dumps

Core Dump

Production Container

```
$ gdb -c core.dump $(which ruby)
```



```
$ gdb -c core.dump $(which ruby)
GNU gdb (Ubuntu 15.0.50.20240403-0ubuntu1) 15.0.50.20240403-git
...
(gdb) bt
#0  __pthread_kill_implementation (no_tid=0, signo=11, threadid=<optimized out>) at ./nptl/pthread_kill.c:44
#1  __pthread_kill_internal (signo=11, threadid=<optimized out>) at ./nptl/pthread_kill.c:78
#2  __GI___pthread_kill (threadid=<optimized out>, signo=signo@entry=11) at ./nptl/pthread_kill.c:89
#3  0x00007ac5fc44527e in __GI_raise (sig=sig@entry=11) at ../sysdeps/posix/raise.c:26
#4  0x00007ac5fca8f6b8 in ruby_default_signal (sig=sig@entry=11) at signal.c:422
#5  0x00007ac5fc85594b in rb_bug_for_fatal_signal (default_sighandler=0x0, sig=sig@entry=11, ctx=ctx@entry=0x63ca4bdeec80,
    fmt=fmt@entry=0x7ac5fcce4dc3 "Segmentation fault at %p") at error.c:1134
#6  0x00007ac5fca8e498 in sigsegv (sig=11, info=0x63ca4bdeedb0, ctx=0x63ca4bdeec80) at signal.c:933
#7  <signal handler called>
#8  RB_INT2FIX (i=<error reading variable: Cannot access memory at address 0x1>) at
../../../include/ruby/internal/arithmetic/long.h:118
#9  rb_int2num_inline (v=<error reading variable: Cannot access memory at address 0x1>) at
../../../include/ruby/internal/arithmetic/int.h:242
#10 rb_fiddle_ptr_aref (argc=1, argv=0x7ac5fc616048, self=<optimized out>) at pointer.c:679
#11 0x00007ac5fcb1b2dd in vm_call_cfunc_with_frame_ (ec=0x63ca4bd64d00, reg_cfp=0x7ac5fc715fa0, calling=<optimized out>,
argc=1, argv=<optimized out>,
    stack_bottom=<optimized out>) at /tmp/ruby-build.20250305095717.10574.iHCYie/ruby-3.4.2/vm_insnhelper.c:3801
#12 0x00007ac5fcb302fc in vm_sendish (method_explorer=<optimized out>, block_handler=<optimized out>, cd=<optimized out>,
reg_cfp=<optimized out>,
    ec=<optimized out>) at /tmp/ruby-build.20250305095717.10574.iHCYie/ruby-3.4.2/vm_callinfo.h:415
--Type <RET> for more, q to quit, c to continue without paging--c
#13 vm_exec_core (ec=0x1, ec@entry=0x63ca4bd64d00) at /tmp/ruby-build.20250305095717.10574.iHCYie/ruby-3.4.2/insns.def:898
#14 0x00007ac5fcb37429 in rb_vm_exec (ec=0x63ca4bd64d00) at vm.c:2595
#15 0x00007ac5fcb491fb in rb_iseq_eval_main (iseq=<optimized out>) at vm.c:2861
#16 0x00007ac5fc937ff5 in rb_ec_exec_node (ec=ec@entry=0x63ca4bd64d00, n=n@entry=0x7ac5e0d73ef0) at eval.c:281
#17 0x00007ac5fc93bc93 in ruby_run_node (n=0x7ac5e0d73ef0) at eval.c:319
#18 0x000063ca3b24f186 in rb_main (argv=0x7fff88c7f128, argc=2) at ./main.c:43
```

Capturing Information About Crashes

Capturing Crash Reports

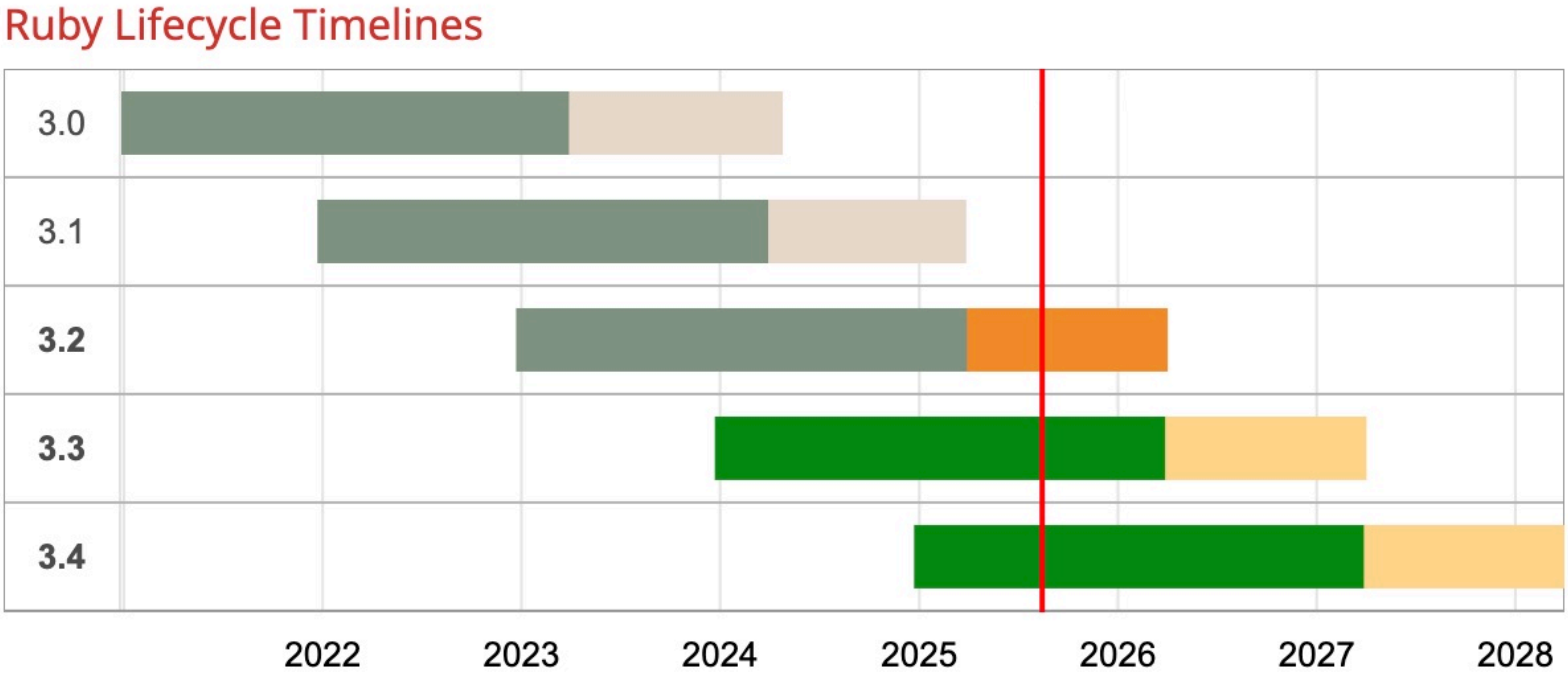
Generating Core Dumps

Debugging Core Dumps

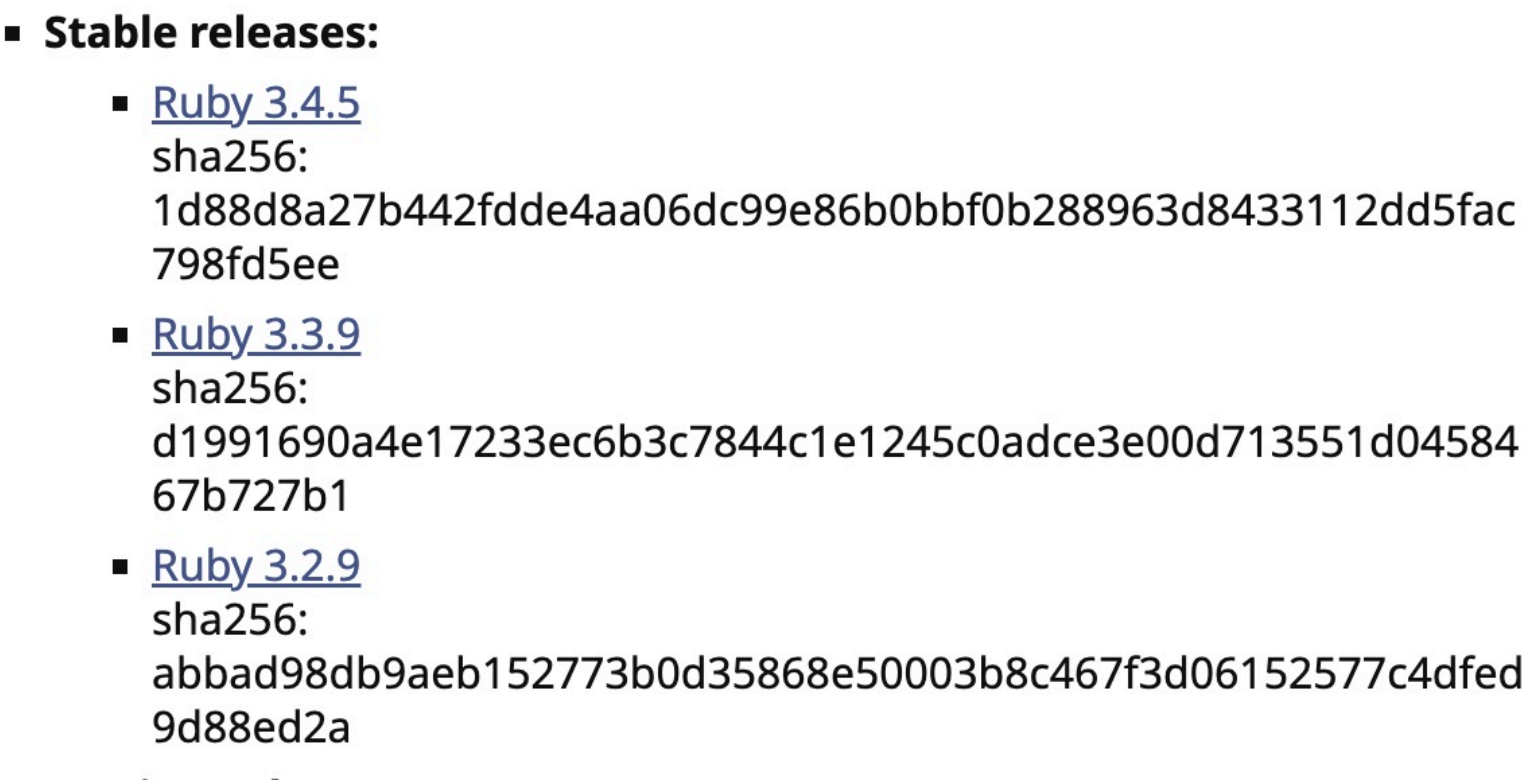
Reporting Bugs

Is Your Ruby Up-to-Date?

www.ruby-lang.org/en/downloads/branches/

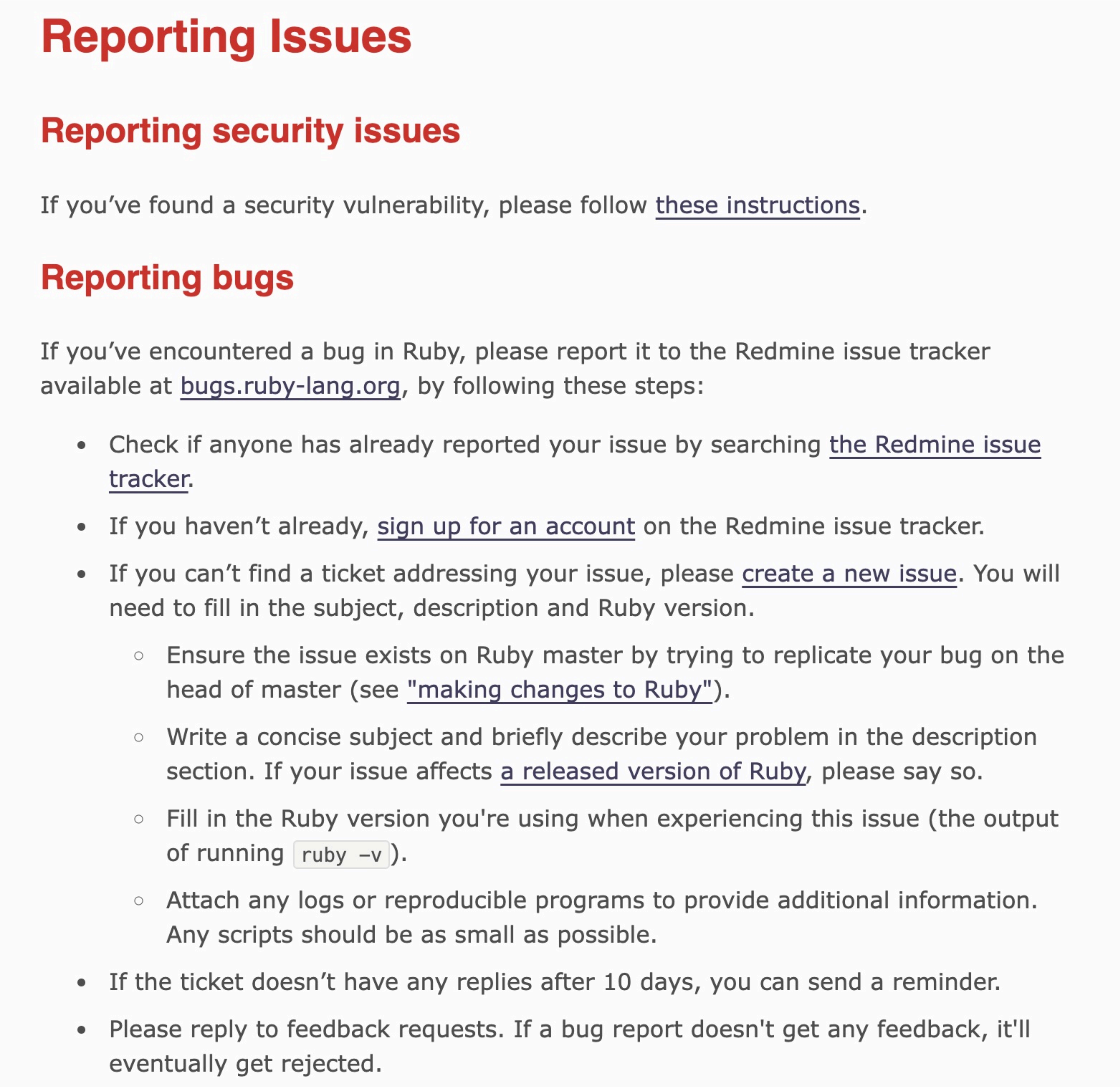


www.ruby-lang.org/en/downloads/



- **Stable releases:**
    - Ruby 3.4.5
      sha256:
      1d88d8a27b442fdde4aa06dc99e86b0bbf0b288963d8433112dd5fac
      798fd5ee
    - Ruby 3.3.9
      sha256:
      d1991690a4e17233ec6b3c7844c1e1245c0adce3e00d713551d04584
      67b727b1
    - Ruby 3.2.9
      sha256:
      abbad98db9aeb152773b0d35868e50003b8c467f3d06152577c4dfed
      9d88ed2a

docs.ruby-lang.org/en/master/contributing/reporting_issues_md.html



Reporting Issues

Reporting security issues

If you’ve found a security vulnerability, please follow these instructions.

Reporting bugs

If you’ve encountered a bug in Ruby, please report it to the Redmine issue tracker available at bugs.ruby-lang.org, by following these steps:

- Check if anyone has already reported your issue by searching the Redmine issue tracker.
- If you haven’t already, sign up for an account on the Redmine issue tracker.
- If you can’t find a ticket addressing your issue, please create a new issue. You will need to fill in the subject, description and Ruby version.
    - Ensure the issue exists on Ruby master by trying to replicate your bug on the head of master (see "making changes to Ruby").
    - Write a concise subject and briefly describe your problem in the description section. If your issue affects a released version of Ruby, please say so.
    - Fill in the Ruby version you're using when experiencing this issue (the output of running `ruby -v`).
    - Attach any logs or reproducible programs to provide additional information. Any scripts should be as small as possible.
- If the ticket doesn’t have any replies after 10 days, you can send a reminder.
- Please reply to feedback requests. If a bug report doesn't get any feedback, it'll eventually get rejected.

Running Patched Versions of Ruby in Production

github.com/Shopify/ruby-definitions

Shopify / ruby-definitions
Public
Notifications
Fork 5
Star 58
Code
Issues
Pull requests
Actions
Projects
Models
Security
Insights
main
7 Branches
4 Tags
Go to file
Code

peterzhu2118 Merge pull request #63 from Shopify/dependabot/github_actio... 4c1c678 · yesterday 146 Commits

.github	Bump actions/checkout from 4 to 5	2 days ago
exe	Fix usage description for "versions" subcommand	2 years ago
lib	Release 1.5.0	last year
rubies	Add `3.4.2-pshopify3`.	5 months ago
test	Check -preview and -pshopify tag order	8 months ago
.gitignore	Centralize Ruby Version to .ruby-version	last year
.rubocop.yml	Centralize Ruby Version to .ruby-version	last year
.ruby-version	Centralize Ruby Version to .ruby-version	last year
CODEOWNERS	Create CODEOWNERS	5 months ago
Gemfile	Initial commit	2 years ago
Gemfile.lock	Bump rexml from 3.2.8 to 3.3.9	5 months ago

About

Definitions to build the Ruby we use at Shopify in development, testing, and production.

- ⛈️ Sources of Instability in Your Infrastructure
- 🐞 Sources of Instability in Ruby
- 🛡️ Preventing Crashes in Production
- 📉 Capturing Information About Crashes

Thank You!

blog.peterzhu.ca

peter@peterzhu.ca

@peterzhu2118

@peterzhu2118@ruby.social

@peterzhu.ca

@peterzhu.photos