Write Ruby extension using Zig

@nishidayuya

2025-04-17

Agileware Drinkup at RubyKaigi 2025 LT

Powered by Rabbit 3.0.5

@nishidayuya (GitHub, X)

- NaCl's programmer. Matz is our fellow.
- from Ruby City Matz-e（松江市）
- now Matz-yama city（松山市）
- i collected RubyKaigi 2024 X posts.
  - from `#rubykaigiA #rubykaigiB #rubykaigiC #rubykaigi #rubykaigi2024`
  - to https://togetter.com/li/2369438
- i will collect RubyKaigi 2025's, too! (posfie)

main issue

Write Ruby extension using Zig

what? Write Ruby extension using Zig

- many Ruby extension using C.
- Rust can write, too:
  - `bundle gem --ext=rust foo`
  - since bundler-3.4.0 (ruby-3.2.0 is bundled)

what’s Zig?

- a programming language.
- Zig can write dynamic link library as C, Rust, and so on.

↓

- Zig can write Ruby extension, I think 🤔

↓

- try it!

today, i will:

- write a method, by Zig code.
- call it.
  - then output return value.

```
ruby -r./zig-out/lib/libzig_ruby_module.so \
  -e 'p(ZigRubyModule.add(1, 2))'
```

step0: run “zig init”

```
$ zig init
info: created build.zig
info: created build.zig.zon
info: created src/main.zig
info: created src/root.zig
info: see `zig build --help` for a menu of options
```

step1: write entrypoint

Init_{so name} function

```
const ruby = @cImport(@cInclude("ruby/ruby.h"));
//...
pub export fn Init_libzig_ruby_module() void {
    const rb_mZigRubyModule =
      ruby.rb_define_module("ZigRubyModule");
    ruby.rb_define_singleton_method(
      rb_mZigRubyModule,
      "add",
      rb_mZigRubyModule_s_add,
      2,
    );
}
```

step2: implement method

```
fn rb_mZigRubyModule_s_add(...) callconv(.c) ruby.VALUE {
    var ap = @cVaStart();
    defer @cVaEnd(&ap);



    _ = @cVaArg(&ap, ruby.VALUE); // dummy
    const o1 = @cVaArg(&ap, ruby.VALUE);
    const o2 = @cVaArg(&ap, ruby.VALUE);
    const n1 = ruby.NUM2INT(o1);
    const n2 = ruby.NUM2INT(o2);
    const sum = add(n1, n2);
    return ruby.INT2NUM(sum);
}
```

step3: fix build rule

```
diff --git a/build.zig b/build.zig
index d8a38d8..62eb5c3 100644
--- a/build.zig
+++ b/build.zig
@@ -48,10 +48,17 @@ pub fn build(b: *std.Build) void {
     // This creates a `std.Build.Step.Compile`, which is the build step responsible
     // for actually invoking the compiler.
     const lib = b.addLibrary(.{
-        .linkage = .static,
+        .linkage = .dynamic,
         .name = "zig_ruby_module",
         .root_module = lib_mod,
     });
+    // ruby -rmkmf -e 'p(RbConfig::CONFIG["libdir"])'
+    lib.addLibraryPath(std.Build.LazyPath{ .cwd_relative = "/home/yuya/.local/share/mise/installs/ruby/3.4.3/lib" });
+    // ruby -rmkmf -e 'p(RbConfig::CONFIG["rubyhdrdir"])'
+    lib.addIncludePath(std.Build.LazyPath{ .cwd_relative = "/home/yuya/.local/share/mise/installs/ruby/3.4.3/include/ruby-3.4.0" });
+    // ruby -rmkmf -e 'p(RbConfig::CONFIG["rubyarchhdrdir"])'
+    lib.addIncludePath(std.Build.LazyPath{ .cwd_relative = "/home/yuya/.local/share/mise/installs/ruby/3.4.3/include/ruby-3.4.0/x86_64-linux" });
+    lib.linkSystemLibrary("c");


     // This declares intent for the library to be installed into the standard
     // location when the user invokes the "install" step (the default step when
```

demo

demo

- `zig init`
- write `Init_libzig_ruby_module`
- implement method
  - `fn rb_mZigRubyModule_s_add(...) callconv(.c) ruby.VALUE {`
- fix build rule
- `ruby -r./zig-out/lib/libzig_ruby_module.so -e 'p(ZigRubyModule.add(1, 2))'`

congrats!

i was writing CFP for RubyKaigi 2025,

, but...

i was writing CFP for RubyKaigi 2025, but...

KATAFRAKT.ME Posts TIL Now!

Long story short: I wrote a Ruby extension in Zig

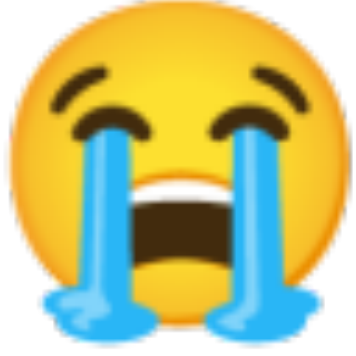
BY PAWEŁ ŚWIĄTKOWSKI
25 DEC 2022

- i found a blog:
  - written by Paweł Świątkowski on 2022-12-25!
  - (zig-0.14.0 (current version) cannot build it.)
    - i sent pull-request to fix it.

i was writing CFP for RubyKaigi 2025, but...

i wrote WHEEL!

i could not write CFP.



Thanks!
by
@nishidayuya