Rust Meetup Zürich

Learning Rust:
Ownership & Borrowing

Sebastian Wicki
@gandro23

Credits:

Niko Matsakis et al.

What is Rust?

Rust is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety.

Why Rust

```ruby
class ::String
  def blank?
    /\A[[:space:]]*\z/ == self
  end
end
```

Ruby

Ruby:
964K iter/sec

Why Rust

```c
static VALUE
rb_str_blank_as(VALUE str)
{
  rb_encoding *enc;
  char *s, *e;

  enc = STR_ENC_GET(str);
  s = RSTRING_PTR(str);
  if (!s || RSTRING_LEN(str) == 0) return Qtrue;

  e = RSTRING_END(str);
  while (s < e) {
    int n;
    unsigned int cc = rb_enc_codepoint_len(s, e, &n, enc);

    switch (cc) {
      case 9:
      case 0xa:
      case 0xb:
      case 0xc:
      case 0xd:
      case 0x20:
      case 0x85:
      case 0xa0:
      case 0x1680:
      case 0x2000:
      case 0x2001:
      case 0x2002:
      case 0x2003:
      case 0x2004:
      case 0x2005:
      case 0x2006:
      case 0x2007:
      case 0x2008:
      case 0x2009:
      case 0x200a:
      case 0x2028:
      case 0x2029:
      case 0x202f:
      case 0x205f:
      case 0x3000:
#if ruby_version_before_2_2()
      case 0x180e:
#endif
          /* found */
          break;
      default:
          return Qfalse;
    }
    s += n;
  }
  return Qtrue;
}
```

C

github.com/SamSaffron/fast_blank

Ruby:
964K iter/sec

C:
10.5M iter/sec

Why Rust

```ruby
class ::String
  def blank?
    /\A[[:space:]]*\z/ == self
  end
end
```

Ruby

```rust
extern “C” fn blank(buf: Buf) -> bool {
  buf.as_slice()   // string slice
     .chars()      // unicode iterator
     .all(|c| c.is_whitespace())
}
```

Rust

Ruby:
964K iter/sec

C:
10.5M iter/sec

Rust:
11M iter/sec

Why Rust

```
fn load_images(paths: &[PathBuf]) -> Vec<Image> {
  paths.iter()
       .map(|path| {
         Image::load(path)
       })
       .collect()
}
```

Why Rust

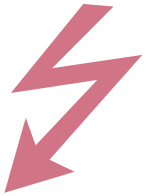
```
extern crate rayon;

fn load_images(paths: &[PathBuf]) -> Vec<Image> {
  paths.par_iter()
       .map(|path| {
         Image::load(path)
       })
       .collect()
}
```

Why Rust

```
extern crate rayon;

fn load_images(paths: &[PathBuf]) -> Vec<Image> {
  let mut jpegs = 0;   // fix: use AtomicU32, Mutex, etc
  paths.par_iter()
       .map(|path| {
         if path.ends_with(“.jpg”) { jpegs += 1; }
         Image::load(path)
       })
       .collect()
}
```



The Rust compiler will statically prevent this data race!

hacking
without fear

Getting Rust

- https://play.rust-lang.org
- https://rustup.rs
    - stable, beta, nightly
- System package manager
- Compile from source

Setting up a Project

We're going to be making a binary project

- the other option is a library.

`cargo new` will create a skeleton project setup for you.

1. **`cd`** to a directory where you like to store code
2. **`cargo new --bin intorust`**
3. **`cd intorust/`**

If you use version control, now would be a good time to commit.

Hello World

```
fn main() {
  println!(“Hello, world!”)
}
```

Learning Material

- **www.rust-tutorials.com/exercises/**
- www.rust-lang.org/documentation.html
- exercism.io
- github.com/ctjhoa/rust-learning