

Introduction to rustc

The Rust Compiler

Boyang Han
yqszxx@gmail.com

11/25/20

Contents

- Rust – The language
- The Rust building infrastructure
- `rustc` – The Rust Compiler
- Current support of RISC-V on Rust

Rust – The language

- A language empowering everyone to build reliable and efficient software.

Why Rust?

Performance

Rust is blazingly fast and memory-efficient: with no runtime or garbage collector, it can power performance-critical services, run on embedded devices, and easily integrate with other languages.

Reliability


Rust's rich type system and ownership model guarantee memory-safety and thread-safety — enabling you to eliminate many classes of bugs at compile-time.

Productivity

Rust has great documentation, a friendly compiler with useful error messages, and top-notch tooling — an integrated package manager and build tool, smart multi-editor support with auto-completion and type inspections, an auto-formatter, and more.

Show me some code!

```
fn main() {  
    println!("Hello, world!");  
}
```



Macro Call

Or this? (for compiler engineers lol)

```
impl CodegenCx<'ll, 'tcx> {  
    pub fn const_array(&self, ty: &'ll Type, elts: &[&'ll Value]) -> &'ll Value {  
        unsafe { llvm::LLVMConstArray(ty, elts.as_ptr(), elts.len() as c_uint) }  
    }  
  
    pub fn const_vector(&self, elts: &[&'ll Value]) -> &'ll Value {  
        unsafe { llvm::LLVMConstVector(elts.as_ptr(), elts.len() as c_uint) }  
    }  
  
    pub fn const_bytes(&self, bytes: &[u8]) -> &'ll Value {  
        bytes_in_context(self.llcx, bytes)  
    }  
  
    ...  
  
}
```

The Rust building infrastructure

- [rustup.rs](#) – The Rust toolchain installer
- [cargo](#) – The Rust package manager
- [crates.io](#) – The Rust Package Registry
- [rustc](#) – The Rust Compiler

rustc – The Rust Compiler

Current support of RISC-V on Rust

Q & A

Thanks!

Boyang Han
yqszxx@gmail.com