Contents

Notes

Medet Can Akuş

May 18, 2019

1 Notes

- By default all rust crates link to standard library, which depends on the operating system for features such as threads, scheduling, file, or networking. It also includes libc library.

- We can disable inclusion of standard library in rust through the no_std attribute.

- Cargo.toml file in a rust project contains project configurations such as dependencies, author, version etc...

- Panic handler function is part of the standard library and it defines the function that the compiler should invoke when the panic occurs. Panic handler is a divergent function.

- Rust has a special syntax for diverging functions which are functions that do not return;

- Language items are special functions and types that are internally required by compiler.

- When you run a rust application first thing that gets executed is not your main function, it is a function that named start. The reason for that because runtime environment needs be set up and exists before your main application. This environment creates the stack places the right arguments on to the register.

- When you turn on your computer it first executes a firmware that reside in your motherboard. This firmware is called bootloader.

- Bootloader firmware in your motherboard is the one who loads your operating system.

- To be able to use inline assembly language you can use asm feature from rust nightly version.

- With rust we can target different platforms or we can specify the target platform for our application with a JSON file. Most fields in the JSON file are required by LLVM.

- Some of the built in libraries in rust such as core, and compiler builtins are compiled for known host triplets and we sometimes need to cross compile it to work with other targets, to do that we can use cargo-xbuild!

- We can specify our target platform on /.cargo/config file without having to pass `--target` argument every time we build our application.

```
{
        "llvm-target": "x86_64-unknown-linux-gnu",
        "data-layout": "e-m:e-i64:64-f80:128-n8:16:32:64-S128",
        "arch": "x86_64",
        "target-endian": "little",
        "target-pointer-width": "64",
        "target-c-int-width": "32",
        "os": "linux",
        "executables": true,
        "linker-flavor": "gcc",
        "pre-link-args": ["-m64"],
        "morestack": false
}
```

- The easiest way to print text into the screen is the VGA Text Buffer

2 References

- **libc**
    - https://www.gnu.org/software/libc/

- **attribute**
    - https://doc.rust-lang.org/rust-by-example/attribute.html

- **panic**
    - https://doc.rust-lang.org/stable/book/ch09-01-unrecoverable-errors-with-panic.html

- **Diverging Function**
    - https://doc.rust-lang.org/1.30.0/book/first-edition/functions.html#diverging-functions

- **Language Items**

- – https://manishearth.github.io/blog/2017/01/11/rust-tidbits-what-is-a-lang-item/
- – https://doc.rust-lang.org/1.5.0/reference.html#language-items
- – https://doc.rust-lang.org/1.5.0/book/lang-items.html

- **Runtime Environment**

  - – https://en.wikipedia.org/wiki/Runtime_system

- **ABI**

  - – https://en.wikipedia.org/wiki/Application_binary_interface

- **Cross Compiler**

  - – https://en.wikipedia.org/wiki/Cross_compiler

- **Bootloader**

  - – https://wiki.osdev.org/Bootloader
  - – https://en.wikipedia.org/wiki/BIOS
  - – https://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface
  - – https://en.wikipedia.org/wiki/GNU_GRUB

- **ASM**

  - – https://doc.rust-lang.org/nightly/unstable-book/language-features/asm.html

- **Specification of Target Platform**

  - – https://llvm.org/docs/LangRef.html#data-layout
  - – https://en.wikipedia.org/wiki/Linker_(computing)
  - – https://lld.llvm.org/
  - – https://os.phil-opp.com/red-zone/
  - – https://en.wikipedia.org/wiki/SIMD
  - – https://os.phil-opp.com/disable-simd/

- **Core and Compiler Builtins**

  - – https://doc.rust-lang.org/nightly/core/index.html

- **Cargo XBuild**

  - – https://github.com/rust-osdev/cargo-xbuild

- **Cargo Config**

  - – https://doc.rust-lang.org/cargo/reference/config.html

- **VGA Text Buffer**

  - – https://en.wikipedia.org/wiki/VGA-compatible_text_mode
  - – https://en.wikipedia.org/wiki/Memory-mapped_I/O