Rust自学

自学前来点自我鼓励

在1月份的时候，在图计算项目的学习任务中曾经学习过一段时间的Rust，但无奈自己当时没有努力去上手，学的时候也是一知半解，现在想想好像和没学一样。后来逐渐感觉到了Rust在系统研究中扮演着很有用的地位，加上自己不知为何对这门语言天然的兴趣，决定逐渐以练手为主，去学习Rust，敲开Rust编程的大门。

Hello World!

- 创建`main.rs`文件，在文件中编程：

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

- 编译、运行：

```
$ rustc main.rs
$ ./main
```

- 输出：

```
Hello Rust!
```

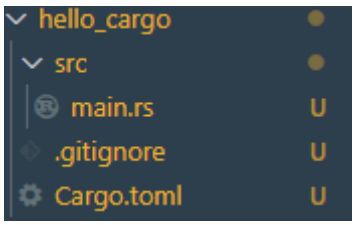
- `println!`：使用`!`意味着在调用一个macro而不是一个normal function.

Hello Cargo!

- Cargo是一个build system & package manager. Cargo会handles很多tasks(比如dependencies这些)，因此常被用来管理Rust project.
- 创建一个cargo项目：

```
$ cargo new hello_cargo
```

- Cargo项目的文件结构：



  - src中是main.rs文件，外面是Cargo.toml：可以理解为配置文件
- Cargo.toml:

```
[package]
name = "hello_cargo"
version = "0.1.0"
edition = "2021"

# See more keys and their definitions at https://doc.rust-
lang.org/cargo/reference/manifest.html

[dependencies]
```

    - name:项目名称（生成可执行文件的名字）
    - version,edition: Rust的版本
    - `[dependencies]` 下面列依赖。
- 编译，运行：

```
$ cargo build
$ ./target/debug/hello_cargo
```

- 输出：

```
Hello, world!
```

- 生成了`cargo.lock`文件，keep tracks of the exact versions of dependencies in project.
- 编译运行一步解决：

```
$ cargo run
```

- 编译但不生成二进制文件（比cargo build更快）：

```
$ cargo check
```

- 优化编译：

```
$ cargo build --release
```