从C++到Rust之路，搭建Rust环境

原创 Ajonbin AJonbin的杂货铺 2023年12月27日 13:58 上海

今天来讲讲怎么搭建Rust环境，创建第一个Rust工程。

安装Rust可以参考官网，按照不同的操作系统，都有详细的安装步骤。

> https://www.rust-lang.org/tools/install

安装完后，会有以下几个重要的命令：

- rustup
- rustc
- cargo

先来看看rustc

rustc是Rust的编译器。但基本上很少直接用到它。有cargo之后，基本都可以通过cargo来管理工程，包括编译，运行，测试等。

最常用的可能就是-V参数，用来看看当前的Rust版本。

```
$ rustc -V
rustc 1.74.1 (a28077b28 2023-12-04)
```


可以看到当前的版本为1.74.1

再来看看cargo

cargo是Rust的包管理工具，有点像JS的npm。 有了它，你可以很方便的创建工程，增加第三方包依赖，编译和运行。

现在让我们用cargo创建第一个rust工程。

运行cargo new <name>来新建一个项目

```
$ cargo new hello_rust
     Created binary (application) `hello_rust` package

$ tree hello_rust/
hello_rust/
├── Cargo.toml
└── src
    └── main.rs

2 directories, 2 files
```


上面的例子中，我们创建了一个新的项目 hello_rust。可以看到cargo会为我们创建：

- hello_rust目录
- hello_rust/Cargo.toml文件，这个是Rust的工程文件
- hello_rust/src/main.rs, 这个就是Rust源文件，以.rs结尾

让我们再来看看Cargo.toml里的内容

```
$ cat hello_rust/Cargo.toml
[package]
name = "hello_rust"
version = "0.1.0"
edition = "2021"

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

[dependencies]
```


里面包含了些基本信息。dependencies里定义了依赖的其他包，可以通过 cargo add <pkg_name> 来添加。

再来看看源代码

```
$ cat hello_rust/src/main.rs
fn main() {
    println!("Hello, world!");
}
```


看着很简单，就是打印一句 hello, world！

最后来通过cargo run命令来运行下

```
$ cd hello_rust/
$ cargo run
   Compiling hello_rust v0.1.0 (/private/tmp/hello_rust)
    Finished dev [unoptimized + debuginfo] target(s) in 3.15s
     Running `target/debug/hello_rust`
Hello, world!
```


先进入hello_rust目录，再运行cargo run，可以看到cargo会先编译再运行。

最后来看看rustup

rustup是用来安装管理Rust工具链的命令。Rust有三个官方发布的版本，叫做channel：

- stable -- 每6周发布一个稳定版本
- beta
- nightly

通过rustup命令可以安装，升级不同的channel，也可以通过rustup切换当前使用的rust版本。

首次安装Rust，默认都是安装的stable版本。

可以通过rustup show命令看看当前的Rust版本

```
$ rustup show
Default host: x86_64-apple-darwin

stable-x86_64-apple-darwin (default)
rustc 1.74.1 (a28077b28 2023-12-04)
```


可以看到默认安装了stable版本。

接着我们来装个beta版本

```
$ rustup toolchain install beta

info: syncing channel updates for 'beta-x86_64-apple-darwin'
info: latest update on 2023-12-22, rust version 1.76.0-beta.1 (0e09125c6 2023-12-21)
info: downloading component 'cargo'
```


再通过rustup show命令查看当前的Rust 版本，可以看到增加了一个channel，但是当前active的仍然是stable版本。

```
$ rustup show

Default host: x86_64-apple-darwin

installed toolchains
--------------------

stable-x86_64-apple-darwin (default)
beta-x86_64-apple-darwin

active toolchain
----------------

stable-x86_64-apple-darwin (default)
rustc 1.74.1 (a28077b28 2023-12-04)
```


现在让我们rustup default 命令来切换到beta版本

```
$ rustup default beta-x86_64-apple-darwin
info: using existing install for 'beta-x86_64-apple-darwin'
info: default toolchain set to 'beta-x86_64-apple-darwin'

  beta-86_64-apple-darwin unchanged - rustc 1.76.0-beta.1 (0e09125c6 2023-12-21)

$ rustup show
Default host: x86_64-apple-darwin

installed toolchains
--------------------

stable-x86_64-apple-darwin
beta-x86_64-apple-darwin (default)

active toolchain
----------------

beta-x86_64-apple-darwin (default)
rustc 1.76.0-beta.1 (0e09125c6 2023-12-21)

$ rustc -V
rustc 1.76.0-beta.1 (0e09125c6 2023-12-21)`
```


可以看到当前Rust版本已经切换成1.76.0beta了。

Rust还在飞速的成长阶段，可以通过rustup update命令定期更新Rust版本，来获取最新的特性。

```
$ rustup update

info: syncing channel updates for 'stable-x86_64-apple-darwin'
info: syncing channel updates for 'beta-x86_64-apple-darwin'
info: checking for self-update

  stable-x86_64-apple-darwin unchanged - rustc 1.74.1 (a28077b28 2023-12-04)
    beta-x86_64-apple-darwin unchanged - rustc 1.76.0-beta.1 (0e09125c6 2023-12-21)

info: cleaning up downloads & tmp directories
```


可以所有已安装的channel都会被更新。

今天就到这里，下次讲讲怎么创建Rust库。