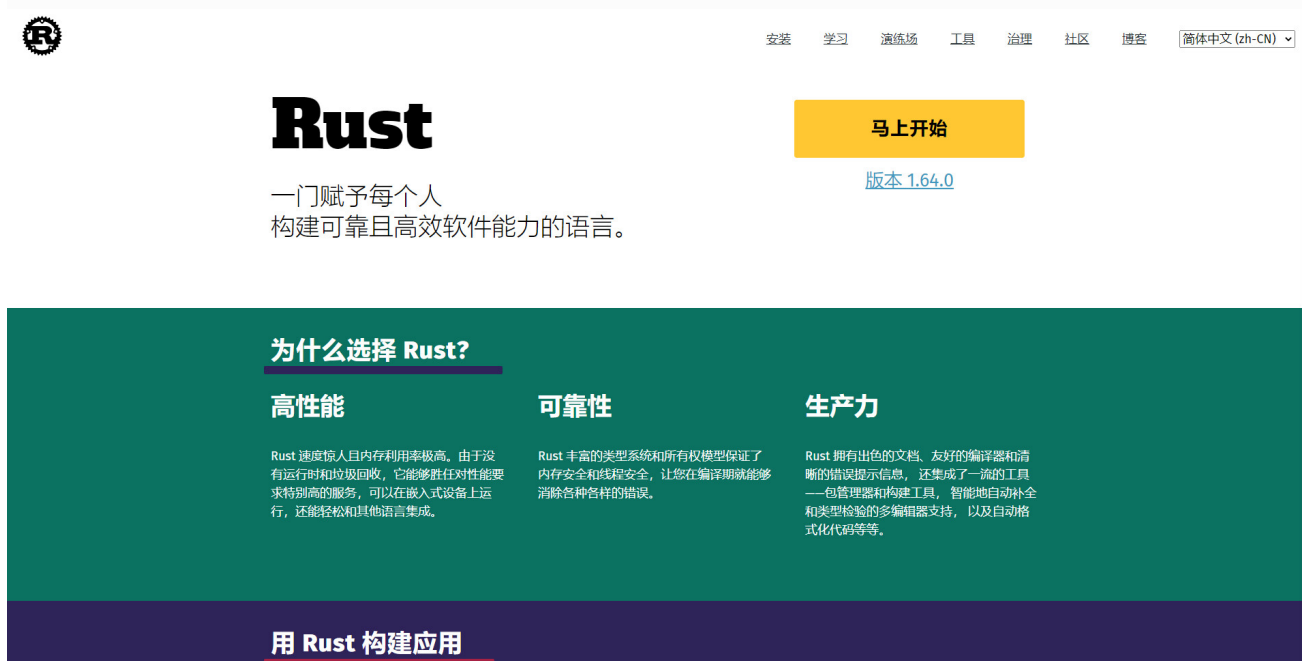


Rust 安装和配置

一、下载

Rust 官网: <https://www.rust-lang.org/zh-CN/>

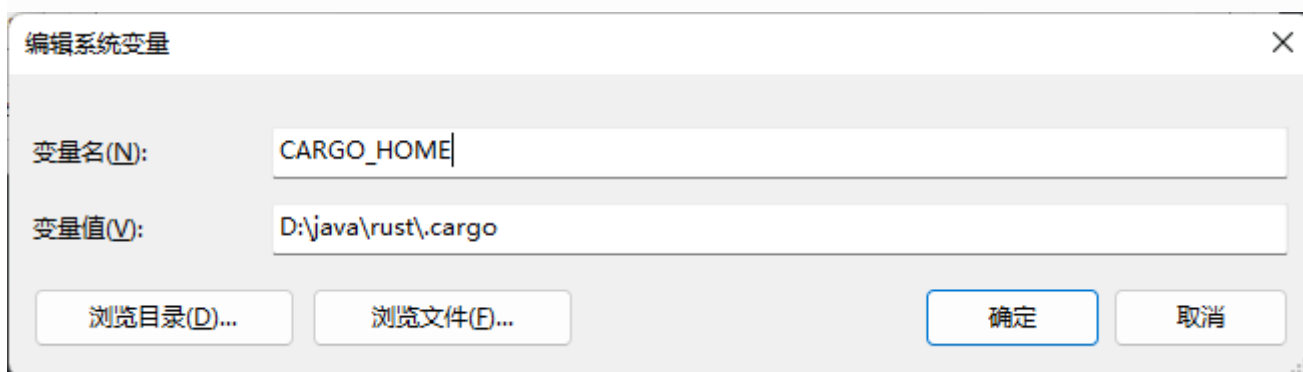
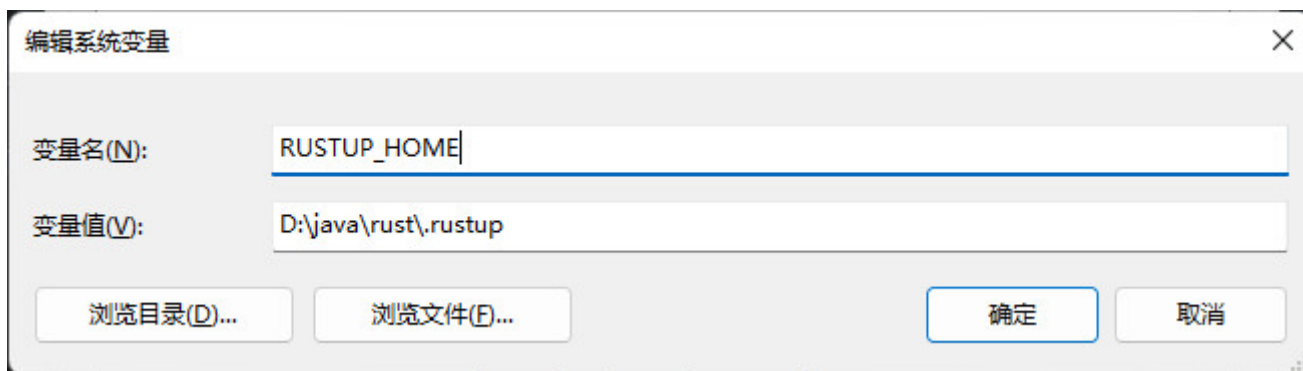


二、改变安装目录

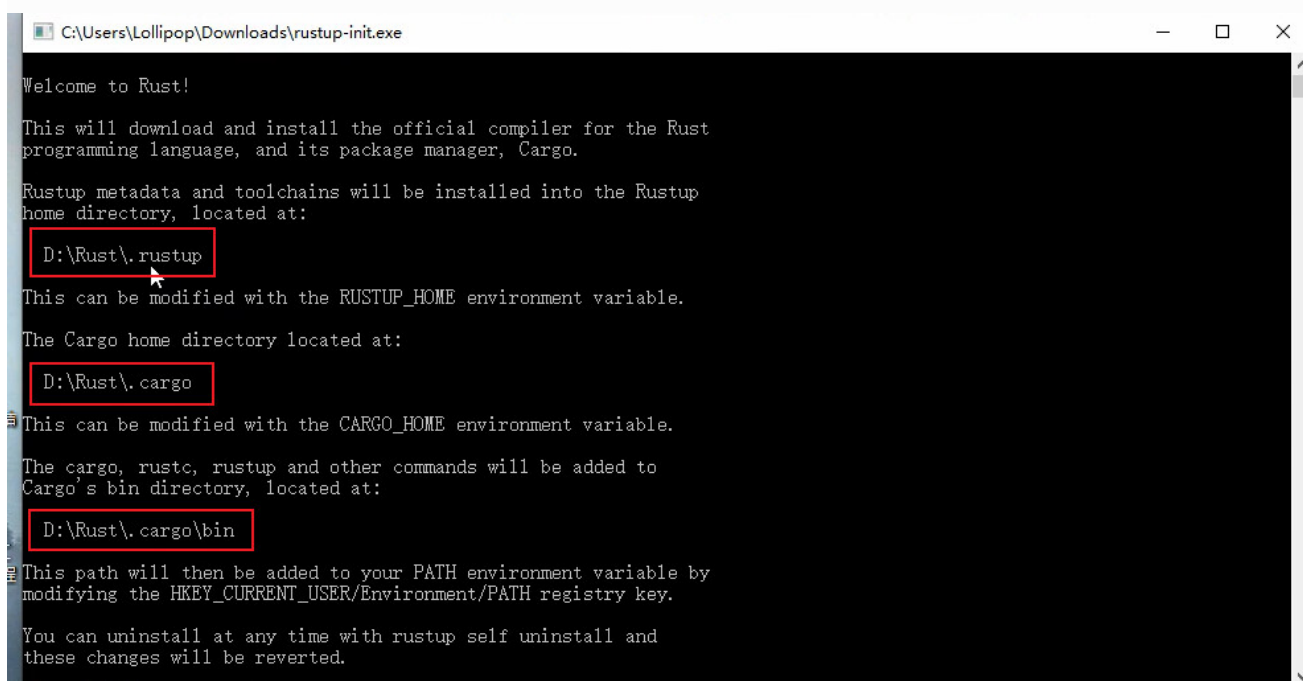
在想安装的位置新建Rust文件夹，并添加 .rustup 和 .cargo 两个文件夹



在系统环境变量中新增两个环境变量，分别指向两个这文件夹

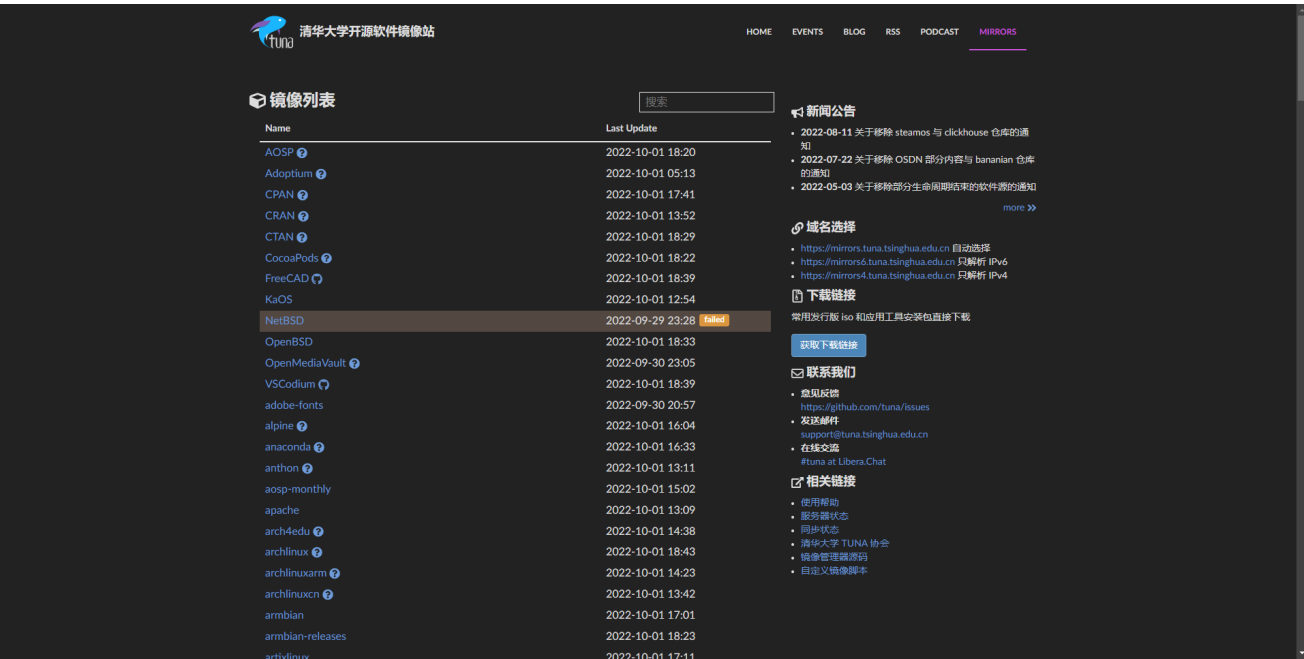


打开从官网下载的安装程序，发现图中三个地方的变成了刚设置的地址

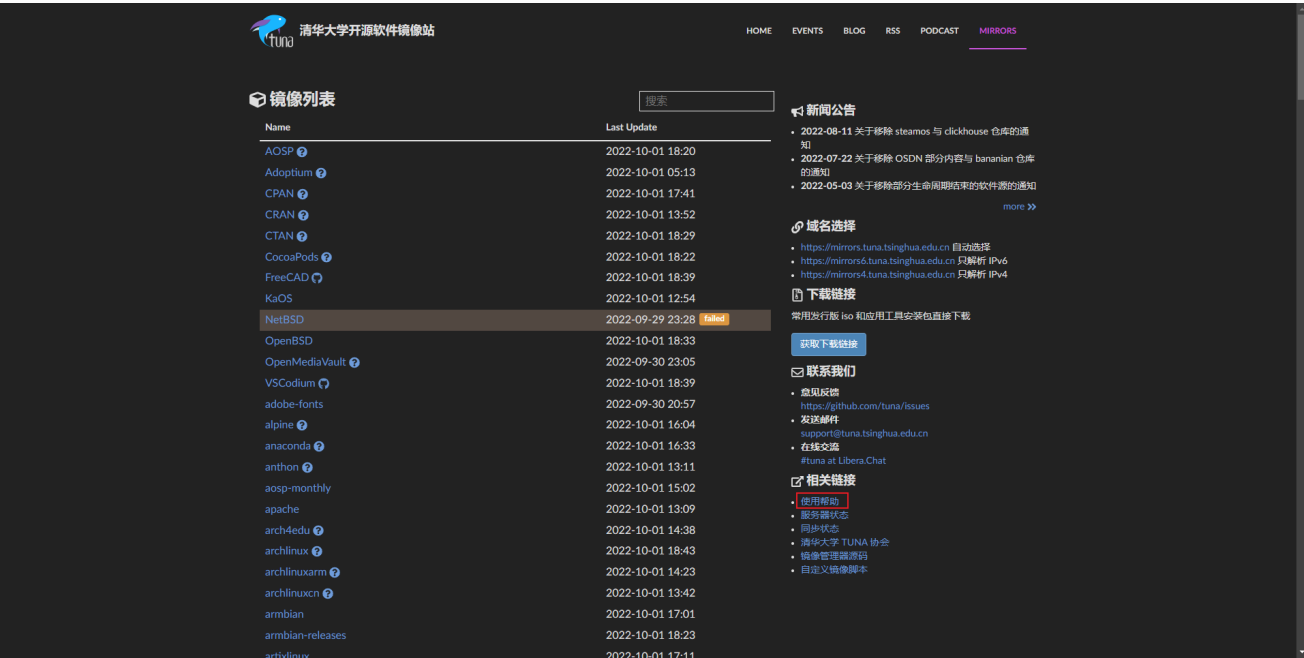


三、配置 rustup 镜像

打开清华大学镜像：<https://mirrors.tuna.tsinghua.edu.cn/>

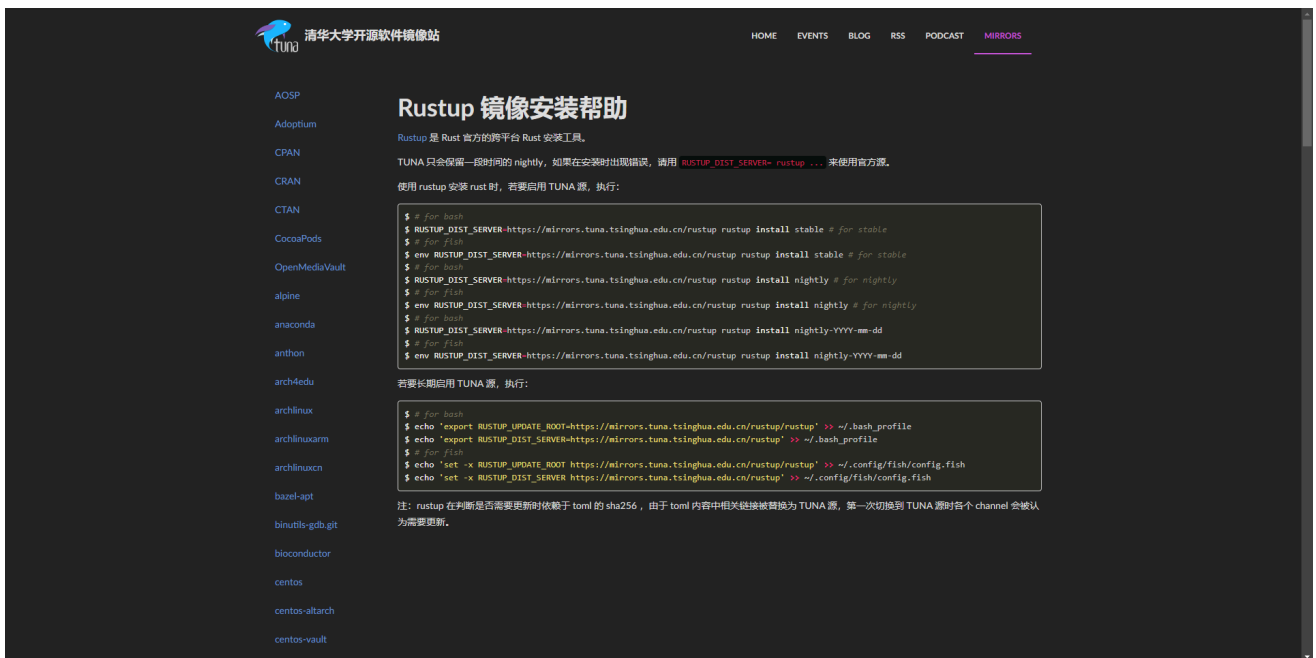


点击右下角的使用帮助

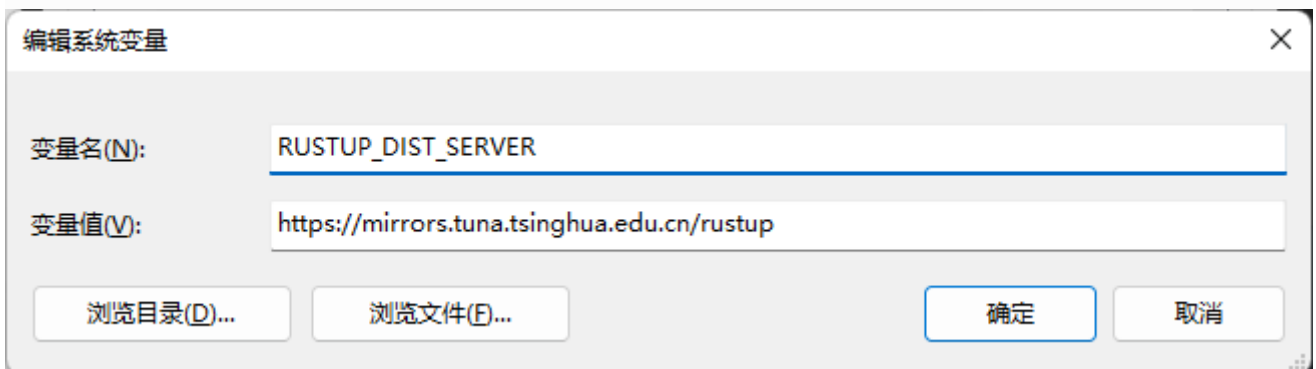
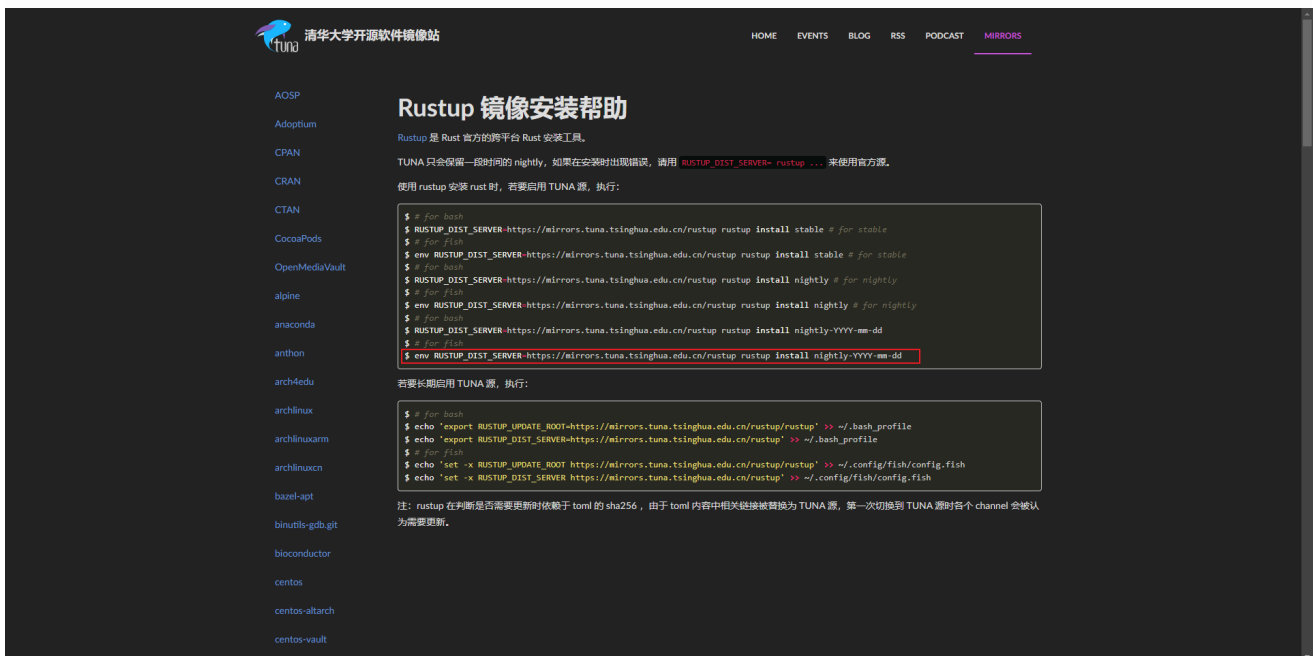


按下 **Ctrl+F** 搜索 rustup，打开如下页面



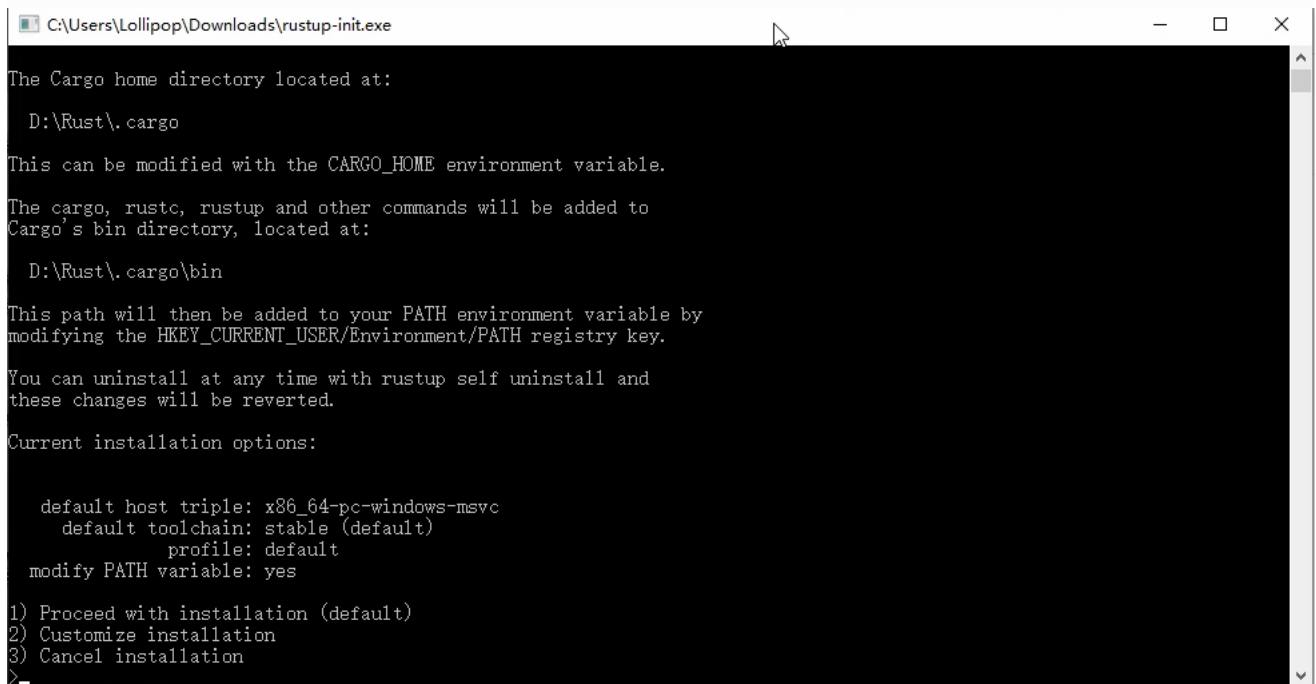


在系统环境变量中新增变量 `RUSTUP_DIST_SERVER=https://mirrors.tuna.tsinghua.edu.cn/rustup`



四、安装

打开安装程序，输入2，回车，进行自定义安装



```
C:\Users\Lollipop\Downloads\rustup-init.exe

The Cargo home directory located at:

  D:\Rust\.cargo

This can be modified with the CARGO_HOME environment variable.

The cargo, rustc, rustup and other commands will be added to
Cargo's bin directory, located at:

  D:\Rust\.cargo\bin

This path will then be added to your PATH environment variable by
modifying the HKEY_CURRENT_USER/Environment/PATH registry key.

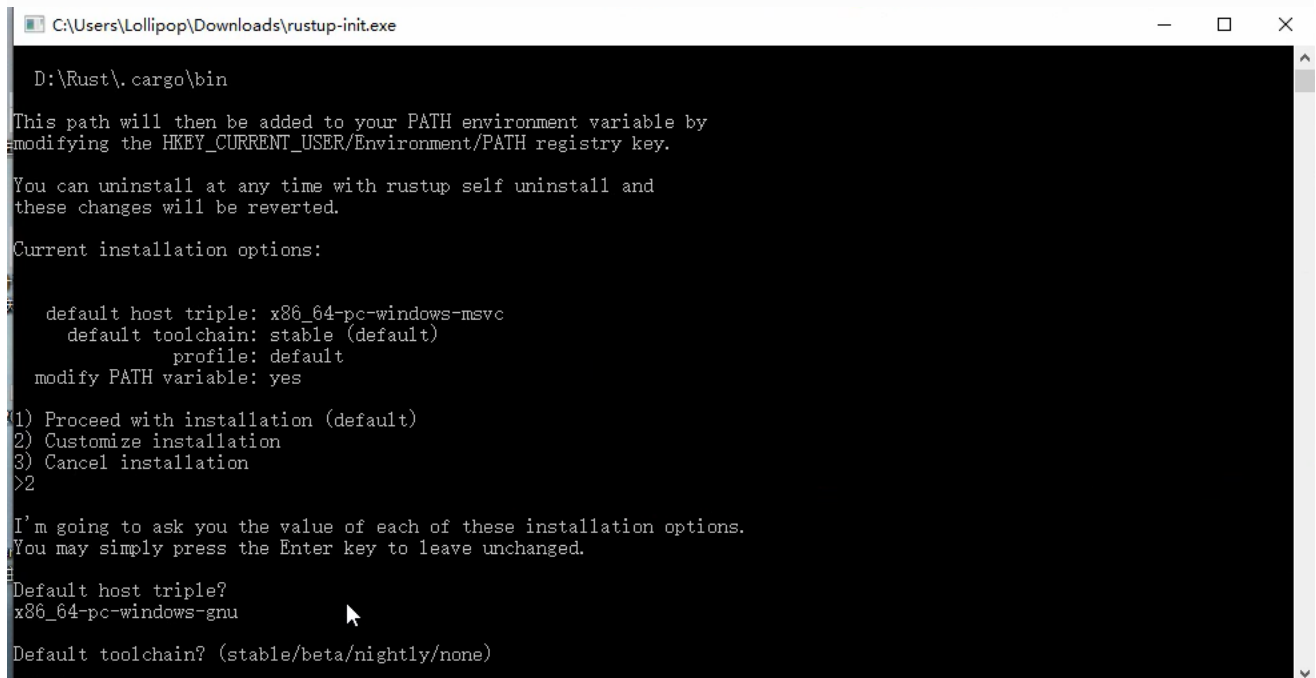
You can uninstall at any time with rustup self uninstall and
these changes will be reverted.

Current installation options:

  default host triple: x86_64-pc-windows-msvc
  default toolchain: stable (default)
  profile: default
  modify PATH variable: yes

1) Proceed with installation (default)
2) Customize installation
3) Cancel installation
>
```

接着输入下列命令，使用 x86_64-pc-windows-gnu



```
C:\Users\Lollipop\Downloads\rustup-init.exe

D:\Rust\.cargo\bin

This path will then be added to your PATH environment variable by
modifying the HKEY_CURRENT_USER/Environment/PATH registry key.

You can uninstall at any time with rustup self uninstall and
these changes will be reverted.

Current installation options:

  default host triple: x86_64-pc-windows-msvc
  default toolchain: stable (default)
  profile: default
  modify PATH variable: yes

1) Proceed with installation (default)
2) Customize installation
3) Cancel installation
>2

I'm going to ask you the value of each of these installation options.
You may simply press the Enter key to leave unchanged.

Default host triple?
x86_64-pc-windows-gnu

Default toolchain? (stable/beta/nightly/none)
```

x86_64-pc-windows-msvc编译速度相对慢一点，因而选用x86_64-pc-windows-gnu，如何选择无伤大雅，不用太纠结。

接着输入 nightly

```
C:\Users\Lollipop\Downloads\rustup-init.exe

D:\Rust\.cargo\bin

This path will then be added to your PATH environment variable by
modifying the HKEY_CURRENT_USER/Environment/PATH registry key.

You can uninstall at any time with rustup self uninstall and
these changes will be reverted.

Current installation options:

    default host triple: x86_64-pc-windows-msvc
    default toolchain: stable (default)
    profile: default
    modify PATH variable: yes

1) Proceed with installation (default)
2) Customize installation
3) Cancel installation
>2

I'm going to ask you the value of each of these installation options.
You may simply press the Enter key to leave unchanged.

Default host triple?
x86_64-pc-windows-gnu

Default toolchain? (stable/beta/nightly/none)
nightly
```

接着直接按回车继续即可

```
C:\Users\Lollipop\Downloads\rustup-init.exe

Modify PATH variable? (y/n)

Current installation options:

    default host triple: x86_64-pc-windows-gnu
    default toolchain: nightly
    profile: default
    modify PATH variable: yes

1) Proceed with installation (default)
2) Customize installation
3) Cancel installation
>

info: profile set to 'default'
info: setting default host triple to x86_64-pc-windows-gnu
info: syncing channel updates for 'nightly-x86_64-pc-windows-gnu'
warning: Signature verification failed for 'https://mirrors.tuna.tsinghua.edu.cn/rustup/dist/channel-rust-nightly.toml'
info: latest update on 2020-08-26, rust version 1.47.0-nightly (bf4342114 2020-08-25)
info: downloading component 'cargo'
info: downloading component 'clippy'
info: downloading component 'rust-docs'
13.0 MiB / 13.0 MiB (100 %) 5.6 MiB/s in 2s ETA: 0s
info: downloading component 'rust-mingw'
info: downloading component 'rust-std'
6.7 MiB / 19.5 MiB (34 %) 0 B/s in 1s ETA: Unknown
```

等待下载安装完成

```
C:\Users\Lollipop\Downloads\rustup-init.exe
info: downloading component 'rust-mingw'
info: downloading component 'rust-std'
19.5 MiB / 19.5 MiB (100 %) 6.6 MiB/s in 2s ETA: 0s
info: downloading component 'rustc'
66.3 MiB / 66.3 MiB (100 %) 5.6 MiB/s in 12s ETA: 0s
info: downloading component 'rustfmt'
info: installing component 'cargo'
info: Defaulting to 500.0 MiB unpack ram
info: installing component 'clippy'
info: installing component 'rust-docs'
13.0 MiB / 13.0 MiB (100 %) 5.2 MiB/s in 2s ETA: 0s
info: installing component 'rust-mingw'
info: installing component 'rust-std'
19.5 MiB / 19.5 MiB (100 %) 13.0 MiB/s in 1s ETA: 0s
info: installing component 'rustc'
66.3 MiB / 66.3 MiB (100 %) 14.2 MiB/s in 4s ETA: 0s
info: installing component 'rustfmt'
info: default toolchain set to 'nightly'

nightly installed - rustc 1.47.0-nightly (bf4342114 2020-08-25)

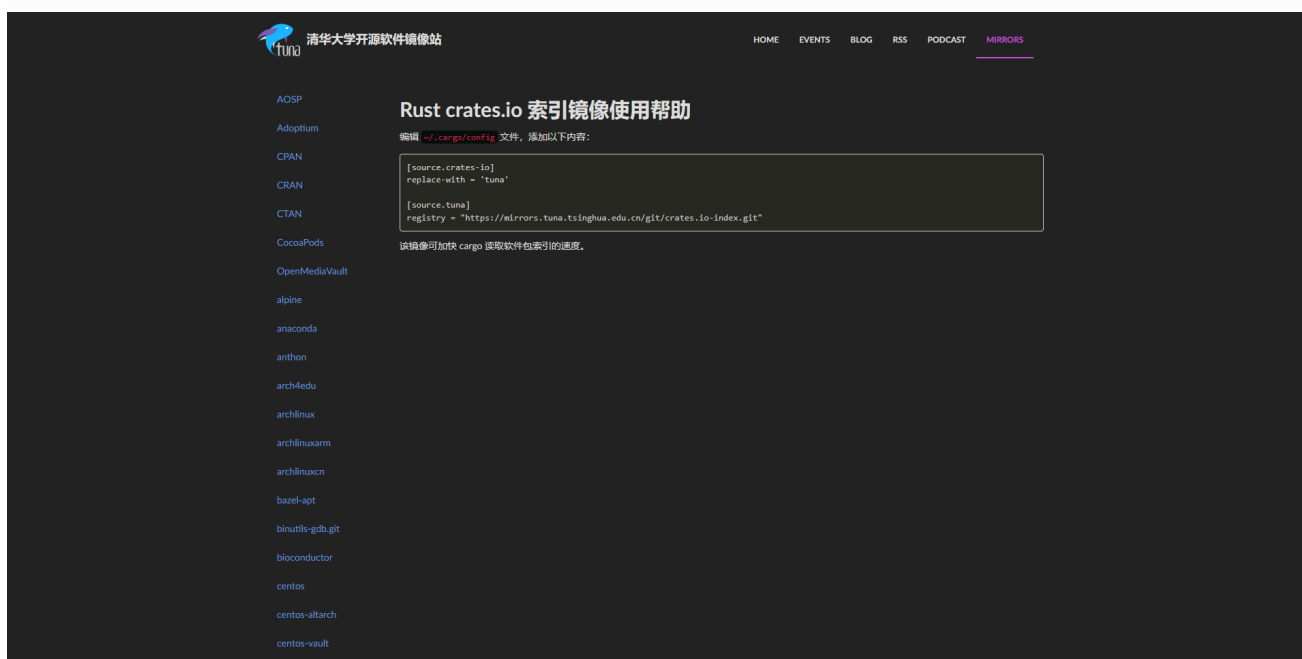
Rust is installed now. Great!

To get started you need Cargo's bin directory (D:\Rust\.cargo\bin) in your PATH
environment variable. Future applications will automatically have the
correct environment, but you may need to restart your current shell.

Press the Enter key to continue.
```

五、配置 crates 镜像

接着在清华镜像站搜索 crates 打开如下页面



在 .cargo 文件夹下新建 config 文件，并添加如下内容

```
1 [source.crates-io]
2 replace-with = 'tuna'
3
4 [source.tuna]
5 registry = "https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git"
```

该镜像可加快 cargo 读取软件包索引的速度。

六、创建工程

在命令行工具中输入如下命令创建rust工程

```
1 cargo new hello-rust
```

接着执行下面的命令运行程序

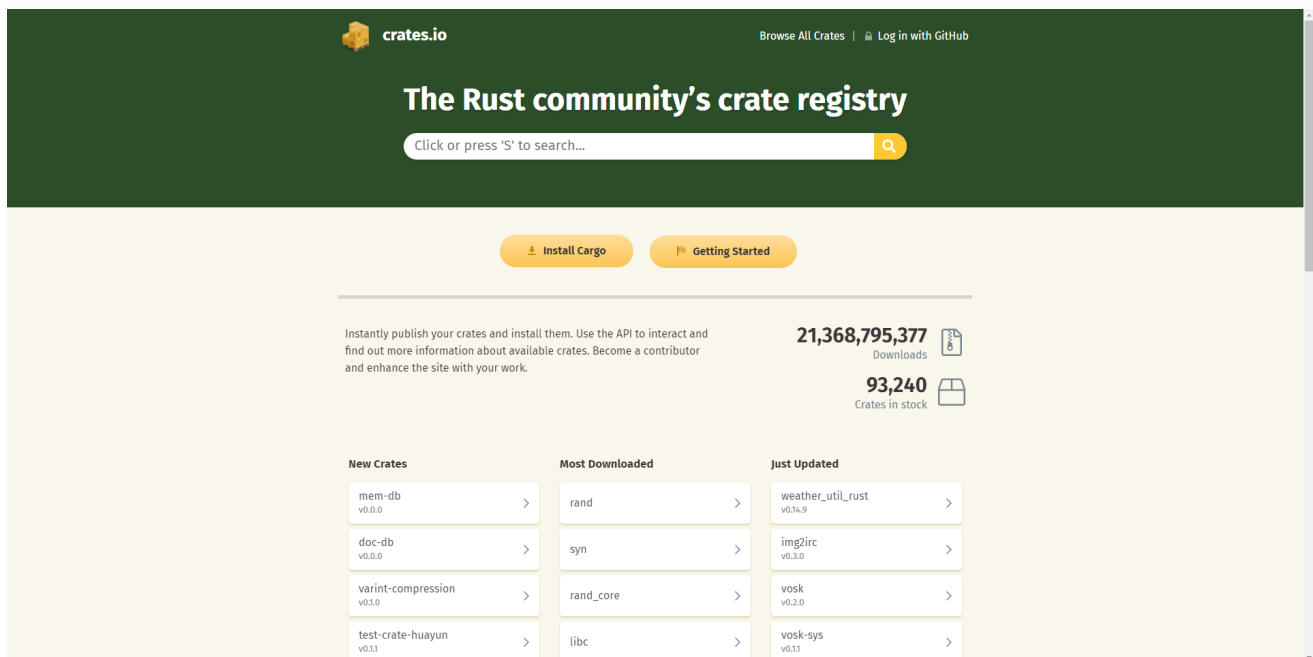
```
1 cargo run
```

执行完后能在终端看到如下内容

```
1 $ cargo run
2   Compiling hello-rust v0.1.0 (/Users/ag_dubs/rust/hello-rust)
3   Finished dev [unoptimized + debuginfo] target(s) in 1.34s
4   Running `target/debug/hello-rust`
5 Hello, world!
```



七、引入依赖

在 `Cargo.toml` 文件中添加以下信息（从 crate 页面上获取<https://crates.io/>）：




The screenshot shows the crates.io website, which is the Rust community's crate registry. The header is dark green with the crates.io logo and navigation links. The main content area is light yellow and features a search bar, two buttons for 'Install Cargo' and 'Getting Started', and statistics on downloads and crates in stock. Below this, there are three columns of crate listings: 'New Crates', 'Most Downloaded', and 'Just Updated'. Each listing shows the crate name, version, and a link to the crate page.

New Crates	Most Downloaded	Just Updated
mem-db v0.0.0	rand	weather_util_rust v0.14.9
doc-db v0.0.0	syn	img2irc v0.3.0
varint-compression v0.1.0	rand_core	vosk v0.2.0
test-crate-huayun v0.1.1	libc	vosk-sys v0.1.1

 crates.io  [Browse All Crates](#) | [Log in with GitHub](#)




Search Results for 'ferris-says'

Displaying 1-4 of 4 total results

Sort by  Relevance ▼




ferris-says v0.2.1
A Rust flavored replacement for the classic cowsay

[Homepage](#) [Documentation](#) [Repository](#)

 All-Time: 175,529
 Recent: 22,490
 Updated: over 1 year ago




ferris-say v0.1.0
Faster Nmap Scanning with Rust

[Homepage](#) [Repository](#)

 All-Time: 161
 Recent: 52
 Updated: 7 months ago




crabsay v0.1.1
A cowsay-like package using ferris-says



[Repository](#)

 All-Time: 202
 Recent: 33
 Updated: 7 months ago

ferris_print v0.1.0
A simple macro to print using ferris say.

[Homepage](#) [Repository](#)

 All-Time: 1,539
 Recent: 179
 Updated: over 4 years ago

 crates.io  [Browse All Crates](#) | [Log in with GitHub](#)

ferris-says v0.2.1

A Rust flavored replacement for the classic cowsay

[#print](#) [#fsays](#) [#ferris](#) [#cowsay](#) [#rustacean](#)

[Readme](#) [5 Versions](#) [Dependencies](#) [Dependents](#)

Ferris Says

A library for printing out text with Ferris as the mascot!

Build requirements

You only need a stable version of the Rust compiler.

How to use the library

Put the following in your `Cargo.toml`:

```
[dependencies]
ferris-says = "0.2"
```

Then import the crate with:




```
extern crate ferris_says;
```

Example

The following bit of code will write the byte string to STDOUT

```
extern crate ferris_says;
```

Metadata

 over 1 year ago
 MIT or Apache-2.0
 11.1 kB

Install

Add the following line to your Cargo.toml file:

```
ferris-says = "0.2.1"
```


Documentation

[docs.rs/ferris-says/0.2.1](#)

Repository

[github.com/mgattozzi/ferris-s...](#)

Owners

 [Michael Gattozzi](#)

Categories

- Command line utilities
- Text processing
- Unicode formatting

Cargo.toml

```
1 [dependencies]
2 ferris-says = "0.2"
```

接着在 **main.rs** 中添加以下代码：

```
1 use ferris_says::say; // from the previous step
2 use std::io::{stdout, BufWriter};
3
4 fn main() {
5     let stdout = stdout();
6     let message = String::from("Hello fellow Rustaceans!");
7     let width = message.chars().count();
8
9     let mut writer = BufWriter::new(stdout.lock());
10    say(message.as_bytes(), width, &mut writer).unwrap();
11 }
```

接着执行 `cargo run` 命令安装依赖

```
PowerShell
https://aka.ms/powershell
Type 'help' to get help.

PS C:\Users\Lollipop> cd D:\Workspaces\Rust
PS D:\Workspaces\Rust> cargo new hello-rust
    Created binary (application) 'hello-rust' package
PS D:\Workspaces\Rust> cd .\hello-rust\
PS D:\Workspaces\Rust\hello-rust> cargo run
    Compiling hello-rust v0.1.0 (D:\Workspaces\Rust\hello-rust)
    Finished dev [unoptimized + debuginfo] target(s) in 4.07s
    Running 'target\debug\hello-rust.exe'
Hello, world!
PS D:\Workspaces\Rust\hello-rust> cargo run
    Updating 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git' index
    Downloaded addr2line v0.13.0 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded adler v0.2.3 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded cfg-if v0.1.10 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded atty v0.2.14 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded rustc-demangle v0.1.16 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded gimli v0.22.0 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded unicode-width v0.1.8 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded error-chain v0.10.0 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded vec_map v0.8.2 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded smallvec v0.4.5 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded clap v2.33.3 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded backtrace v0.3.50 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded libc v0.2.76 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded miniz_oxide v0.4.1 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloaded textwrap v0.11.0 (registry 'https://mirrors.tuna.tsinghua.edu.cn/git/crates.io-index.git')
    Downloading 6 crates, remaining bytes: 1037.0 KB
```