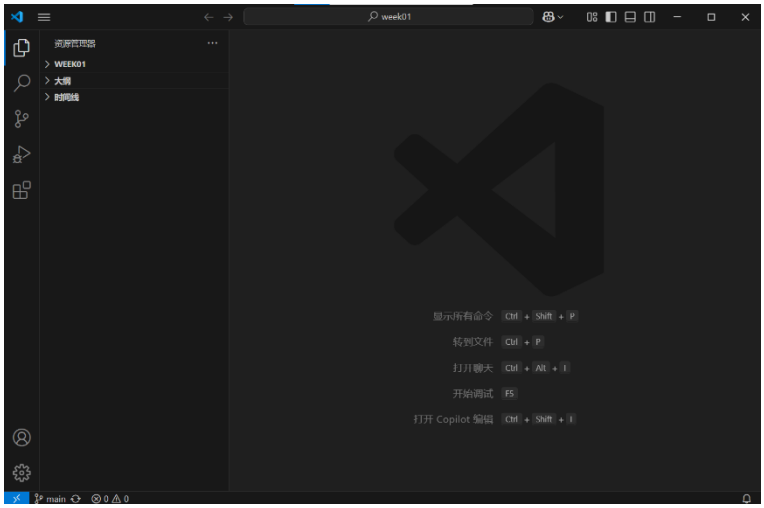
金融编程与计算 第一周作业---准备开发环境

Part 1. 安装 VS Code 和 Git

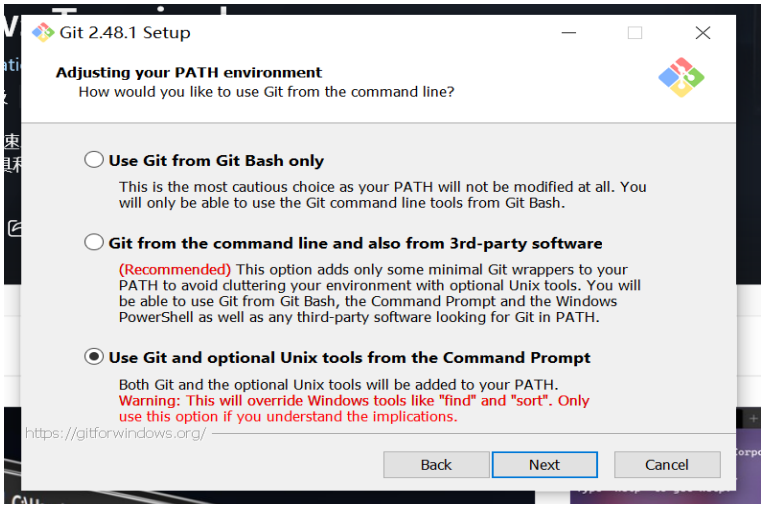
1.之前早已安装 Vscode(已完成)

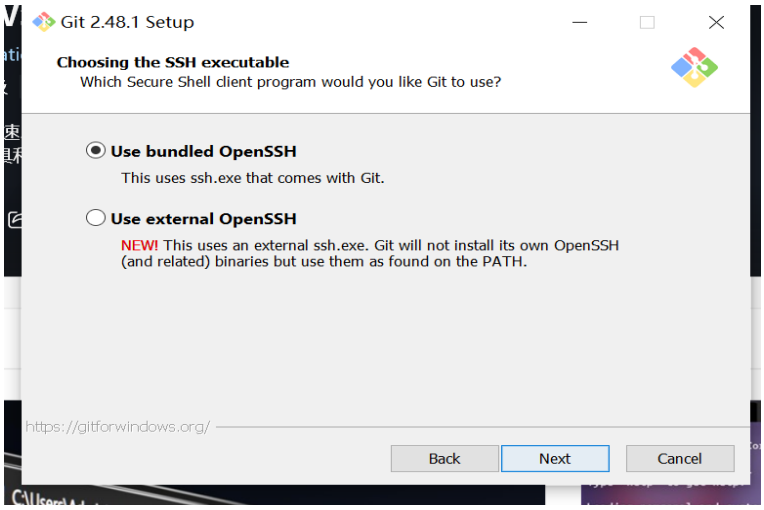


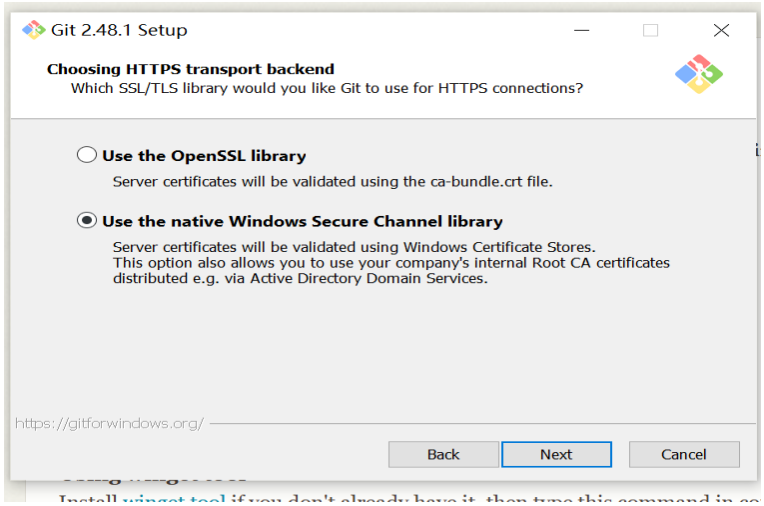
2.安装 Git（按照以下步骤已完成）

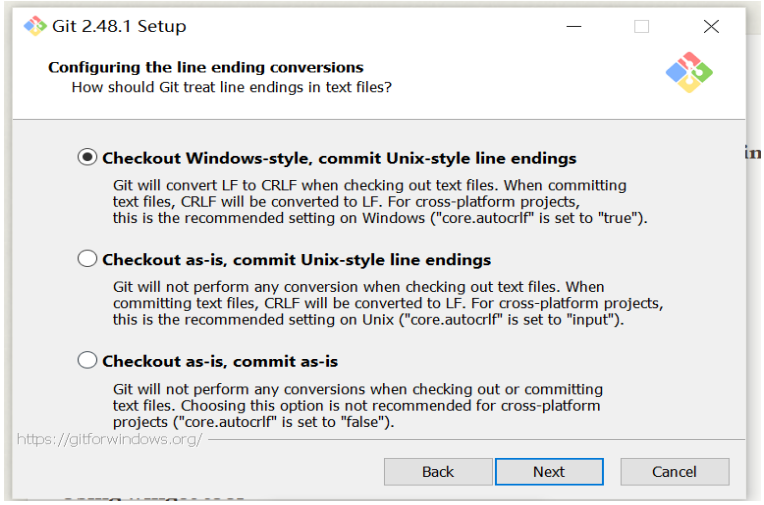
先在微软商店进行终端（terminal）的安装，使用终端和 git 会更容易操作。

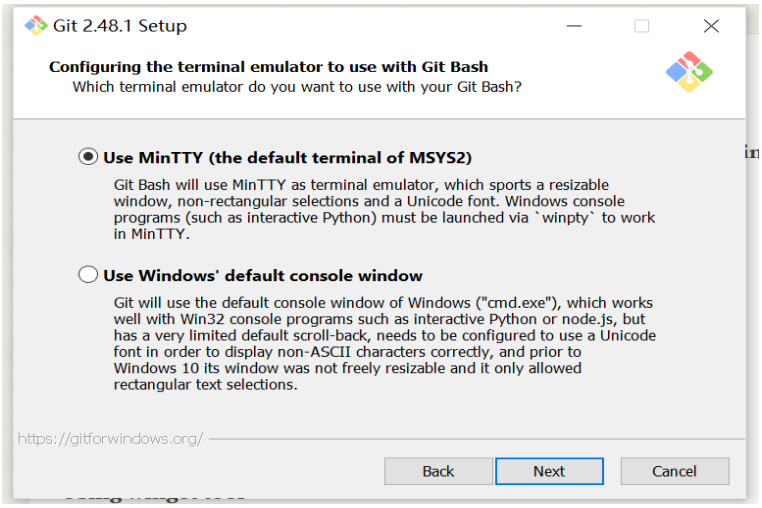
接下来是容易出错的步骤，要格外注意！！！

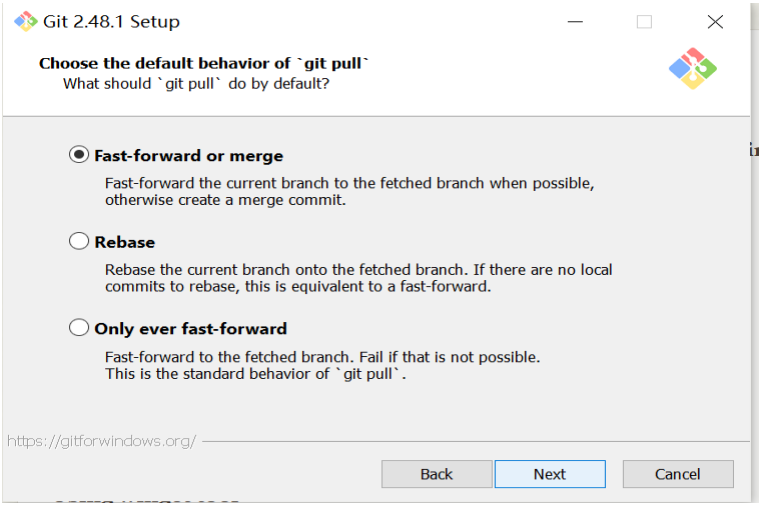


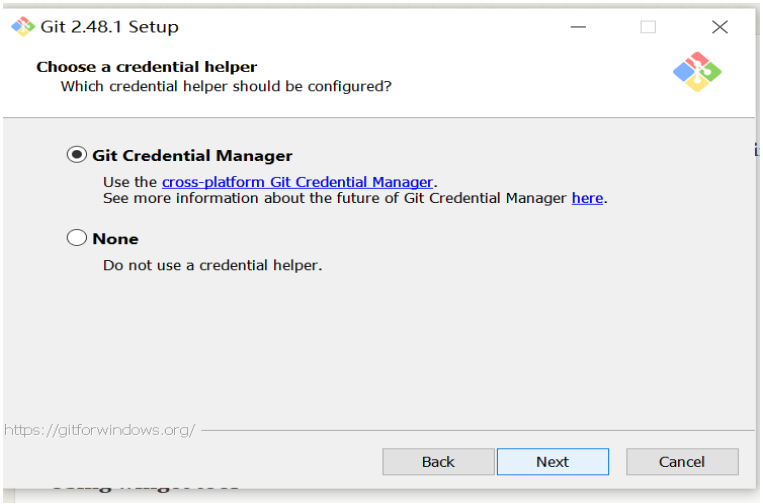

Git 2.48.1 Setup
Choosing the SSH executable
Which Secure Shell client program would you like Git to use?
Use bundled OpenSSH
This uses ssh.exe that comes with Git.
Use external OpenSSH
NEW! This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.
https://gitforwindows.org/
Back
Next
Cancel

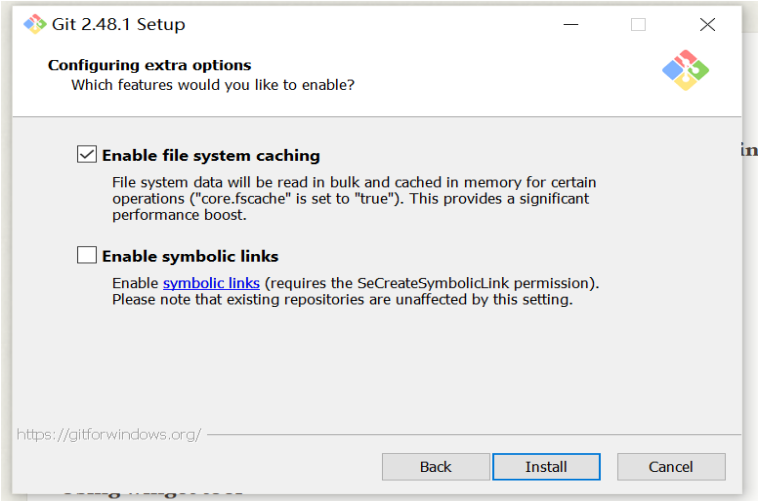

Git 2.48.1 Setup
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?
Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.
Use the native Windows Secure Channel library
Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.
https://gitforwindows.org/
Back
Next
Cancel

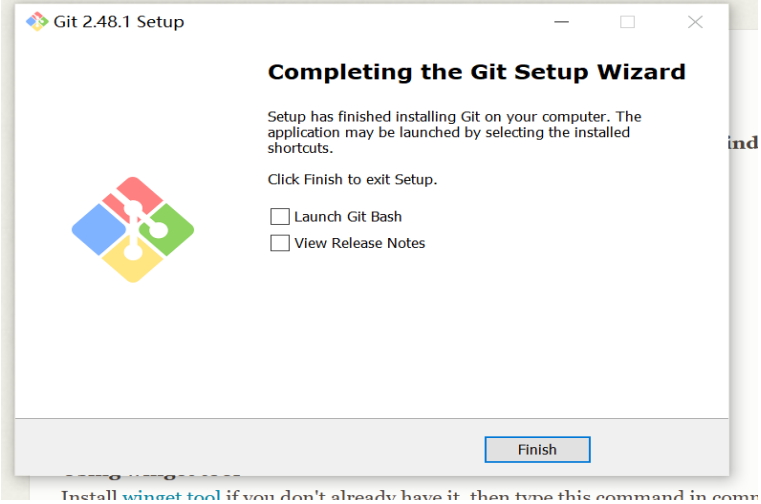

Git 2.48.1 Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Checkout as-is, commit as-is
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").
https://gitforwindows.org/
Back
Next
Cancel


Git 2.48.1 Setup
Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?
Use MinTTY (the default terminal of MSYS2)
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.
Use Windows' default console window
Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.
https://gitforwindows.org/
Back
Next
Cancel

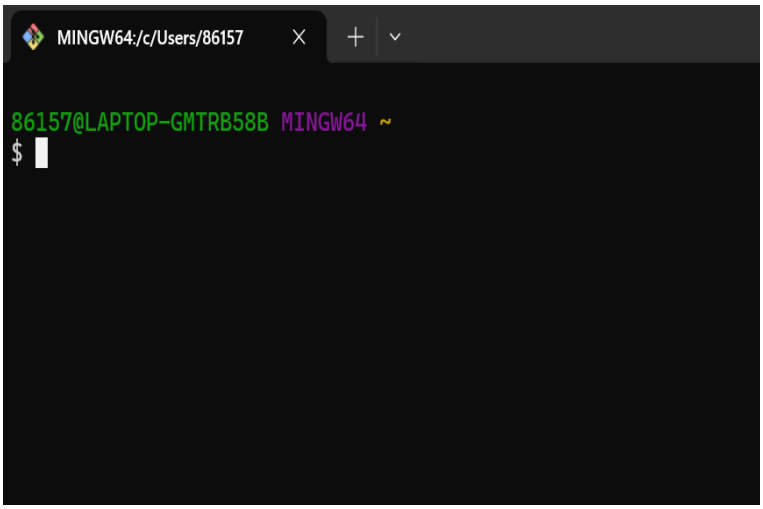

Git 2.48.1 Setup
Choose the default behavior of `git pull`
What should `git pull` do by default?
Fast-forward or merge
Fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.
Rebase
Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.
Only ever fast-forward
Fast-forward to the fetched branch. Fail if that is not possible. This is the standard behavior of `git pull`.
https://gitforwindows.org/
Back
Next
Cancel


Git 2.48.1 Setup
Choose a credential helper
Which credential helper should be configured?
Git Credential Manager
Use the cross-platform Git Credential Manager.
See more information about the future of Git Credential Manager here.
None
Do not use a credential helper.
https://gitforwindows.org/
Back
Next
Cancel

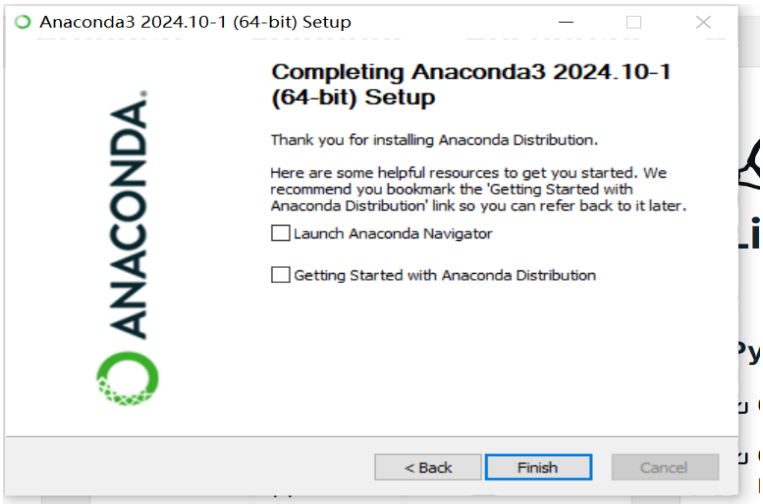


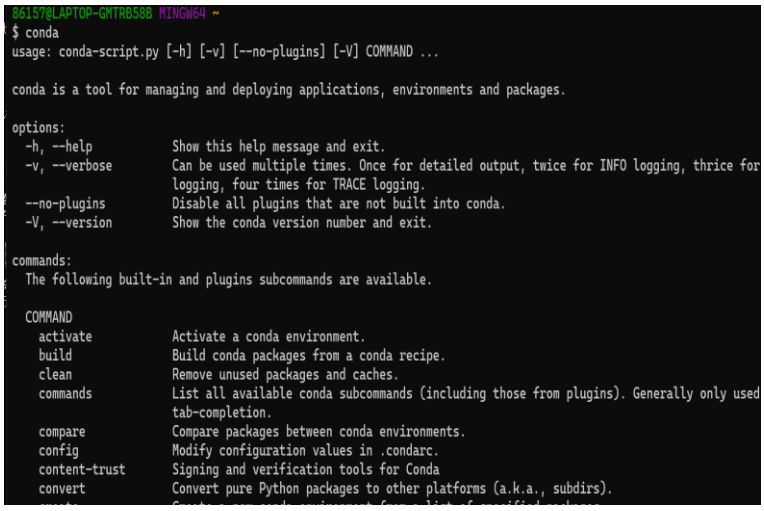


将终端的默认配置文件改为 git，结果如下图：



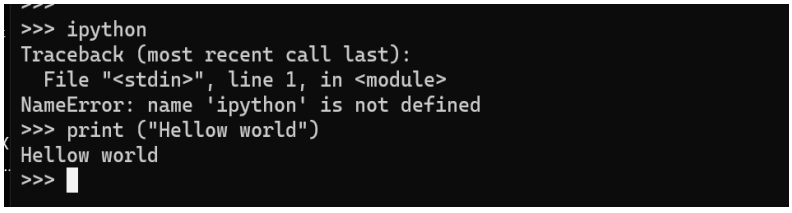
Part 2. 安装 Anaconda（已完成）





Part 3. 使用豆包与设置 SSH 密钥

1.输入简单口令：



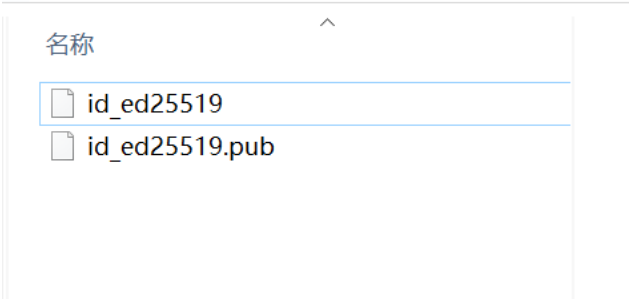
2.生成自己设备的 SSH 密钥（公钥-私钥对）

粘贴命令改为自己的邮箱：

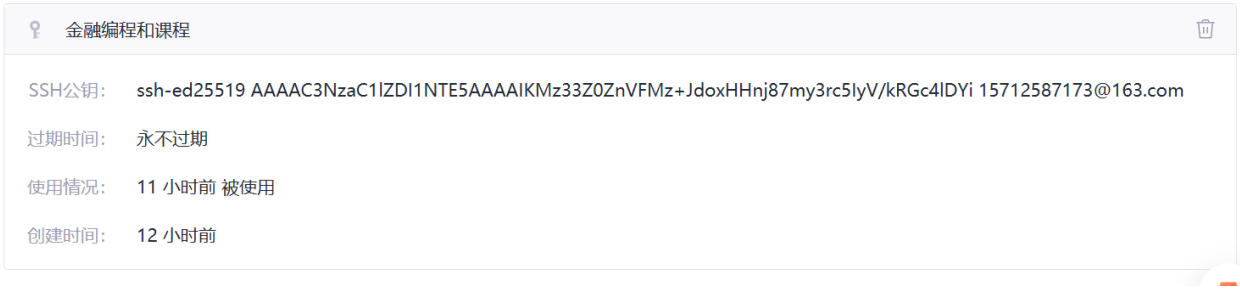
```
86157@LAPTOP-GMTRB58B MINGW64 ~
$ ssh-keygen -t ed25519 -C "15712587173@163.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/86157/.ssh/id_ed25519):
Created directory '/c/Users/86157/.ssh'.
Enter passphrase for "/c/Users/86157/.ssh/id_ed25519" (empty for no passphrase
Enter same passphrase again:
Your identification has been saved in /c/Users/86157/.ssh/id_ed25519
Your public key has been saved in /c/Users/86157/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:KAAgytT8/BAlOA9LEB64hMWapX2svUlrlQVGoLc28kQ 15712587173@163.com
The key's randomart image is:
+--[ED25519 256]--+
|BB* o+o.         |
|Oo+O .+          |
|oX+.E...         |
|= .=o* ..        |
|  .** +oS        |
|  .=ooo.         |
|   ..=           |
|    =            |
|   .             |
+----[SHA256]-----+

86157@LAPTOP-GMTRB58B MINGW64 ~
$
```

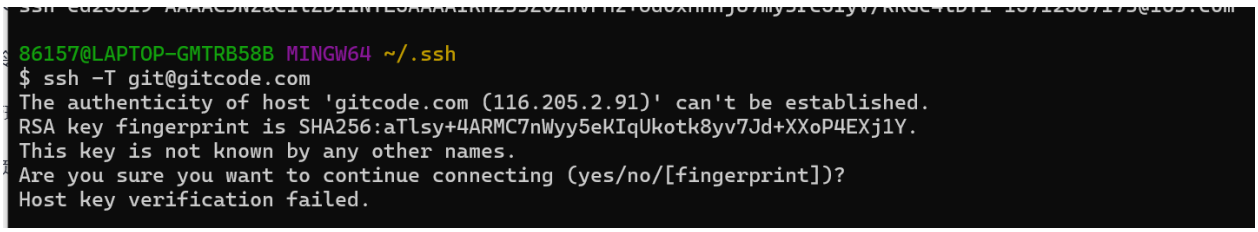
3. 创建密钥后 C 盘会多一个 ssh(C:\Users\ASUS)的文件夹，不同的设备生成不同的密钥。



4.将公钥添加到自己的 GitCode 安全设置里



5.测试是否正确添加了 SSH 密钥，结果如下，结果是已添加

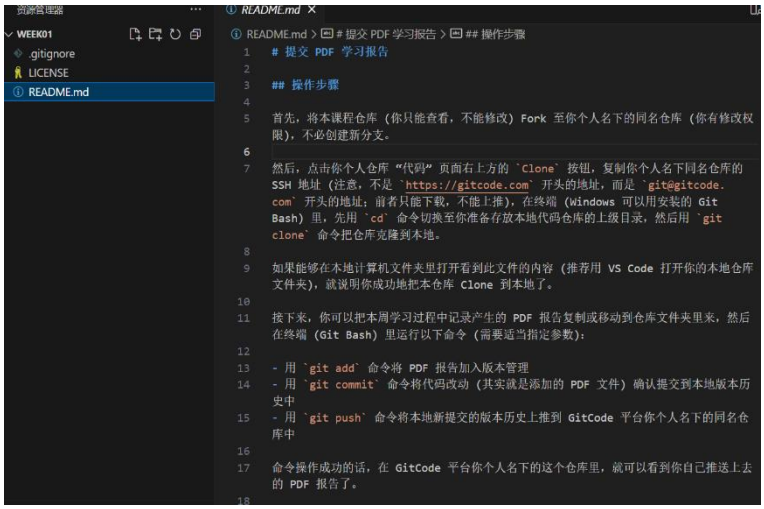


```
86157@LAPTOP-GMTRB58B MINGW64 ~/.ssh
$ ssh -T git@gitcode.com
The authenticity of host 'gitcode.com (116.205.2.91)' can't be established.
RSA key fingerprint is SHA256:aTlsy+4ARMC7nWyy5eKIqUkotk8yv7Jd+XXoP4EXj1Y.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Host key verification failed.
```

Part 4. Git-Fork-PR 作业提交

1. Fork 完成

2. Git clone 克隆在本地 week01 并且在 vs code 打开



3. 作业提交