

# 第一周作业

## 一、遇到的问题

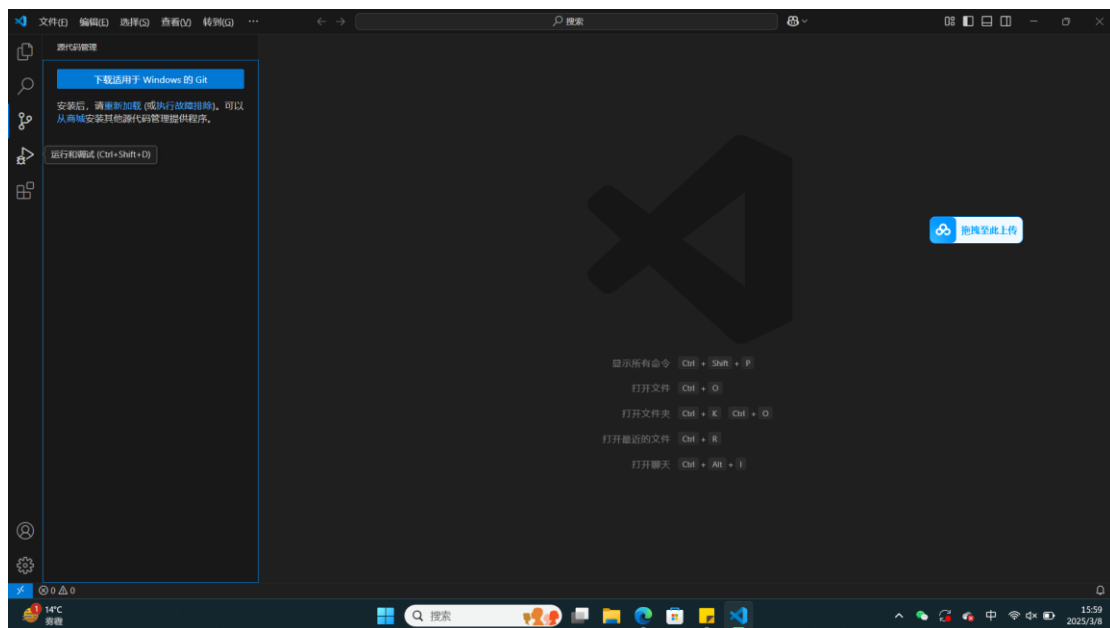
1. 在 window 终端无法打开 Git Bash（使用豆包分析问题，根据他提出的解决办法，解决了问题，详细过程记录在下面）**关键问题是软件的安装路径中包含非英文字符！吸取教训！以后一定要注意类似问题**
2. 创建 SSH 时遇到问题 也是因为中文的用户名 使用豆包解决了问题

### 方法一：手动创建目录

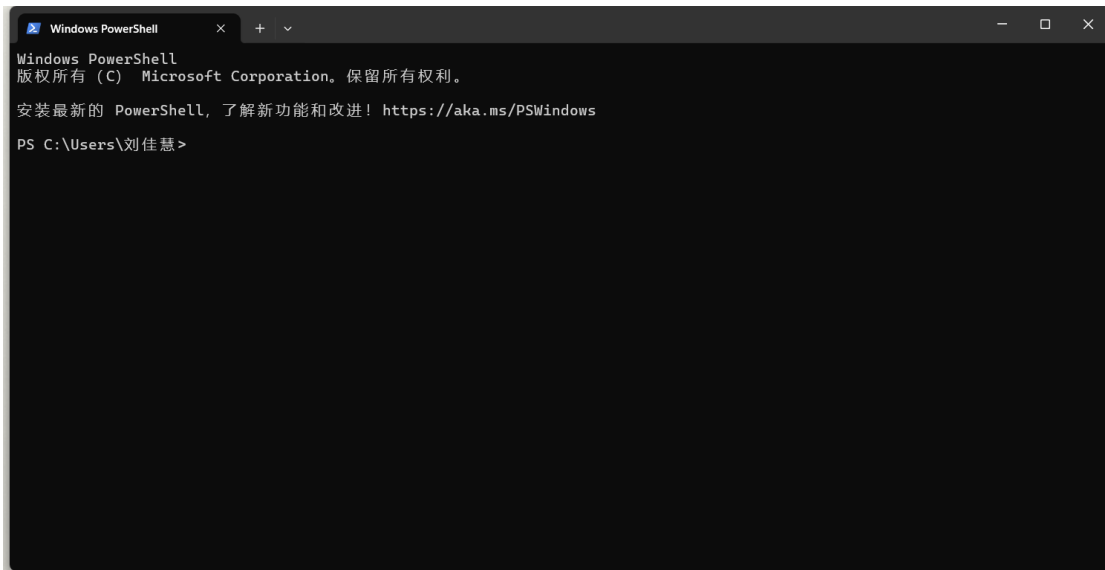
1. 打开文件资源管理器，导航到 `C:\Users\你的用户名` 目录（这里是你当前报错显示的路径部分，虽然乱码但能看出是 `Users` 目录下）。
2. 在该目录下手动创建一个名为 `.ssh` 的文件夹，确保文件夹权限设置为当前用户可读写。
3. 回到命令行窗口，重新运行 `ssh-keygen -t ed25519 -C "你的邮箱"` 命令，按照提示输入即可，此时应该不会再报无法创建目录的错误。

## 二、详细笔记

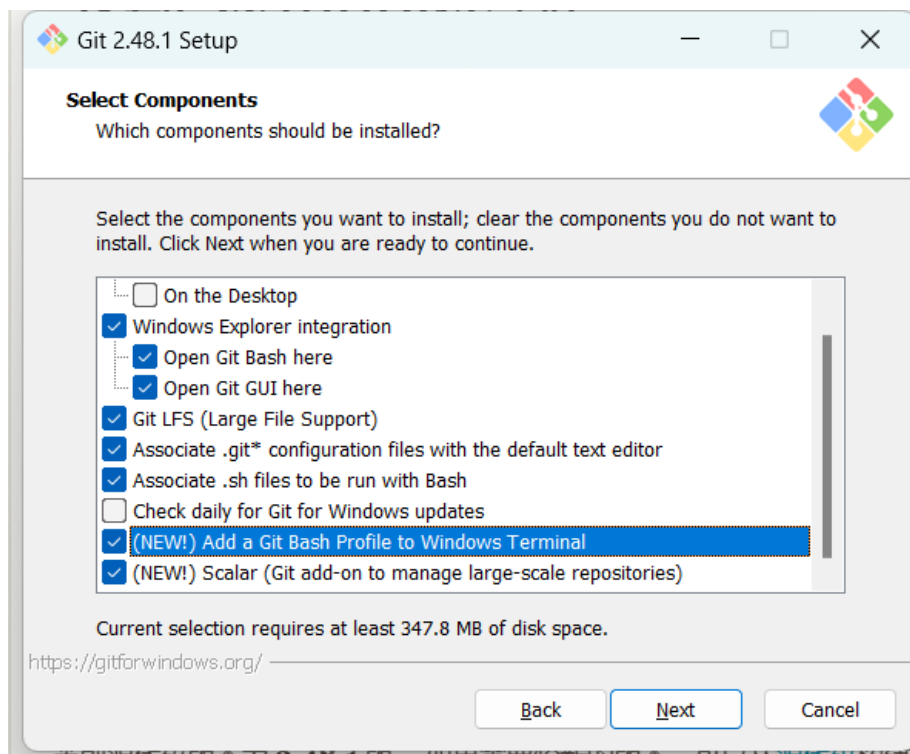
### 1.根据老师的操作方式 完成 VS code 的安装与配置



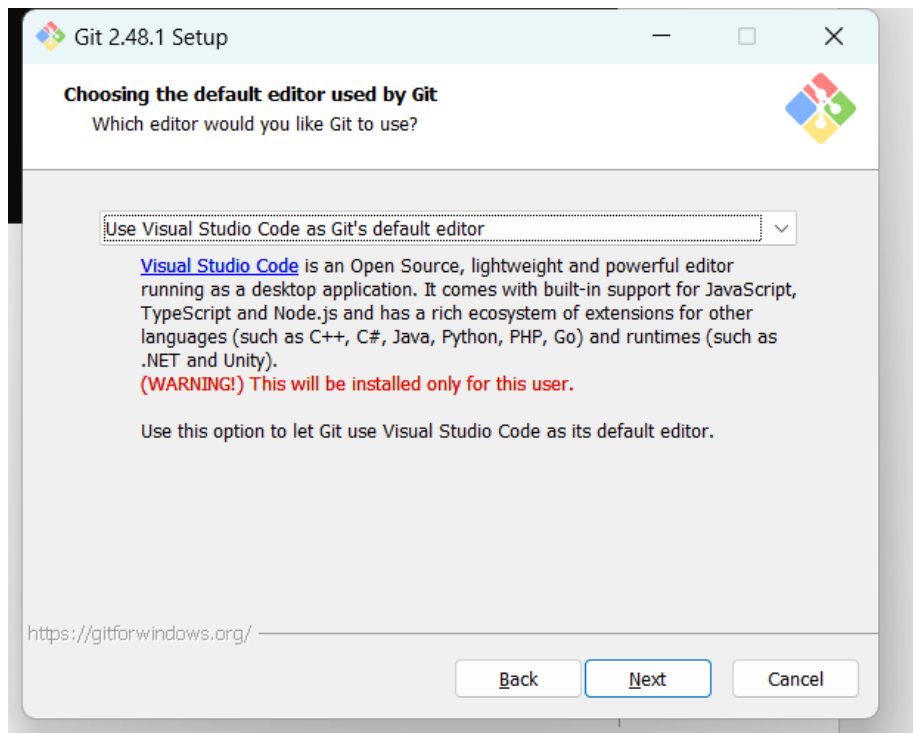
### 安装终端并固定到任务栏中



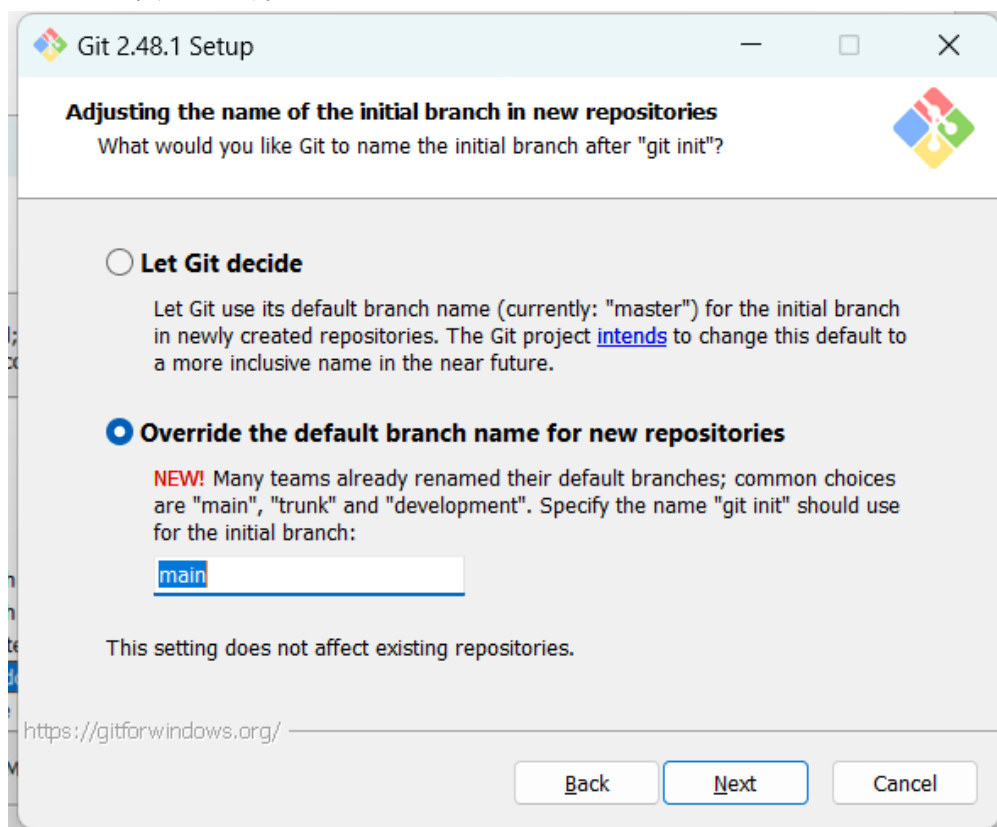
## 2. 下载 git



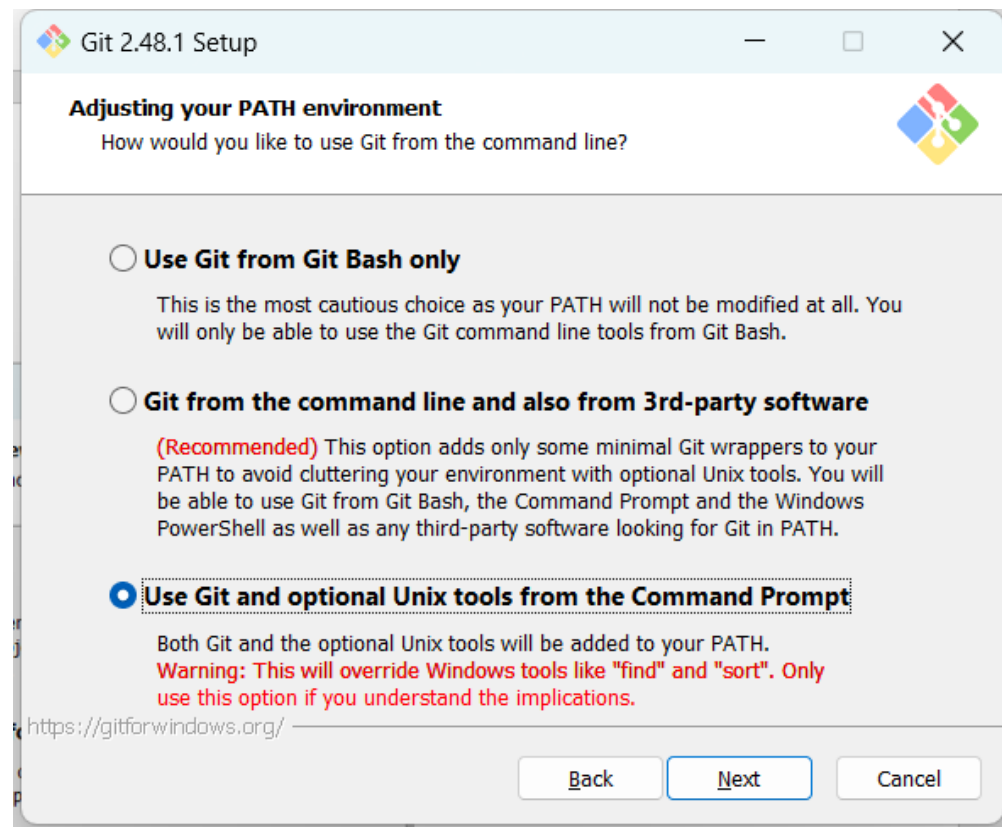
注意：选择 VS code 作为编辑器



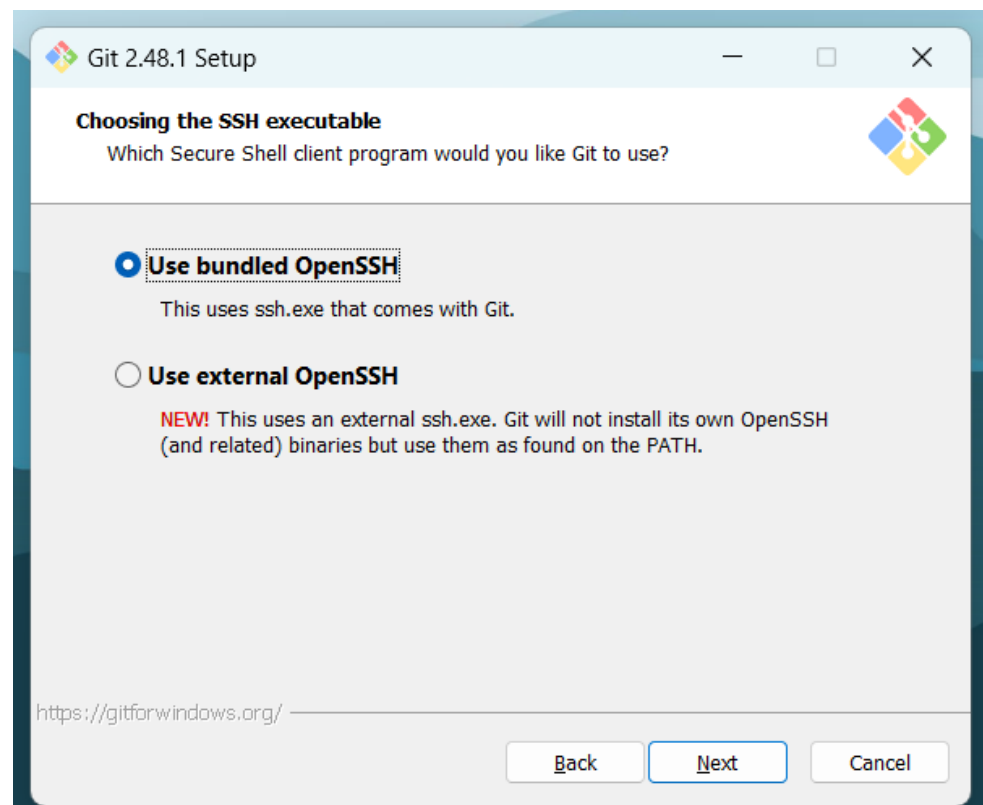
注意：选择以 main 为分支



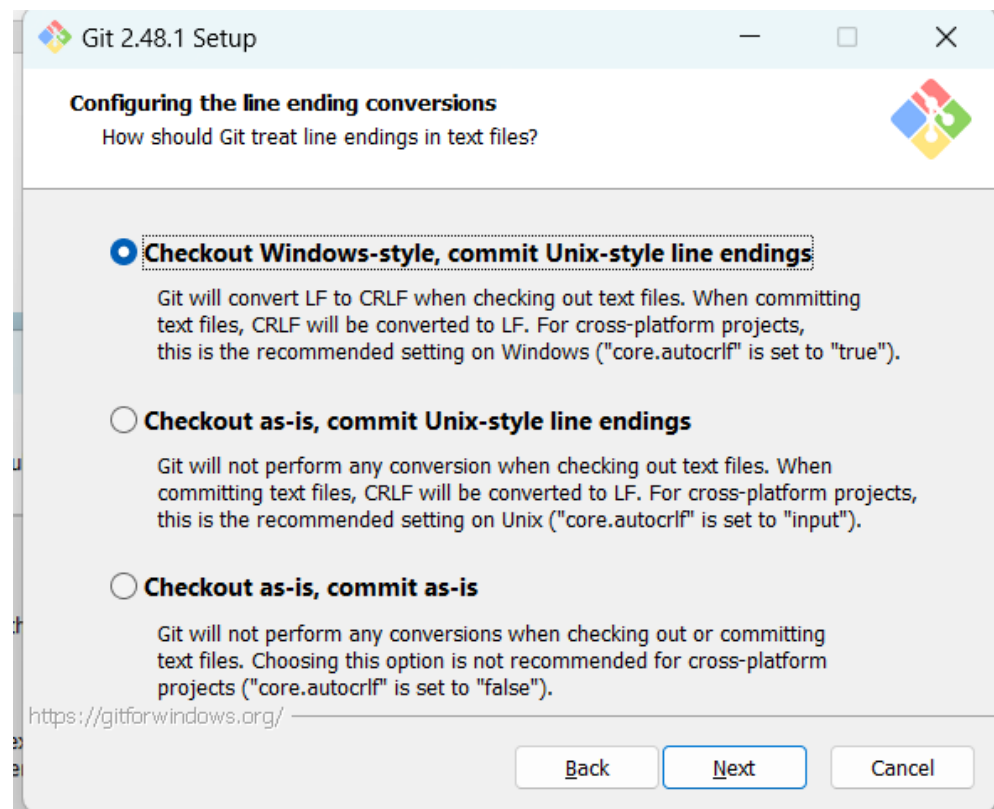
## 配置环境变量 PATH



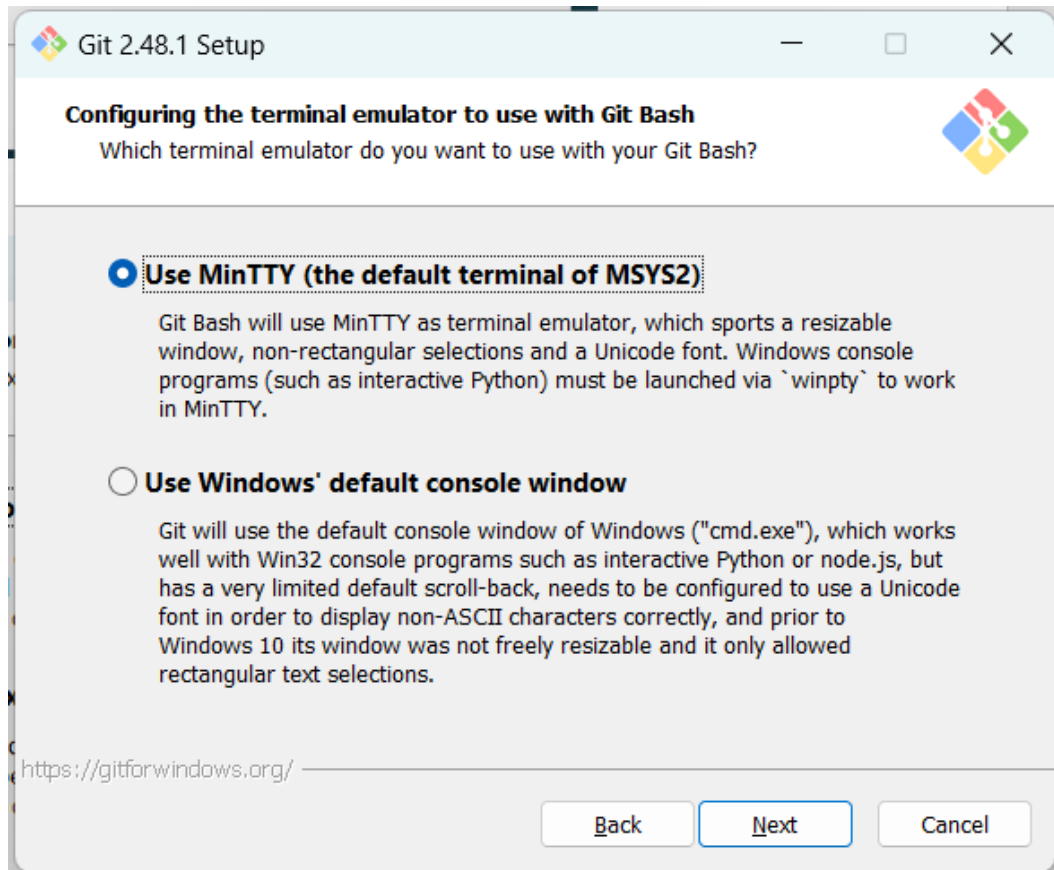
## 加密（选择捆绑的 SSH）



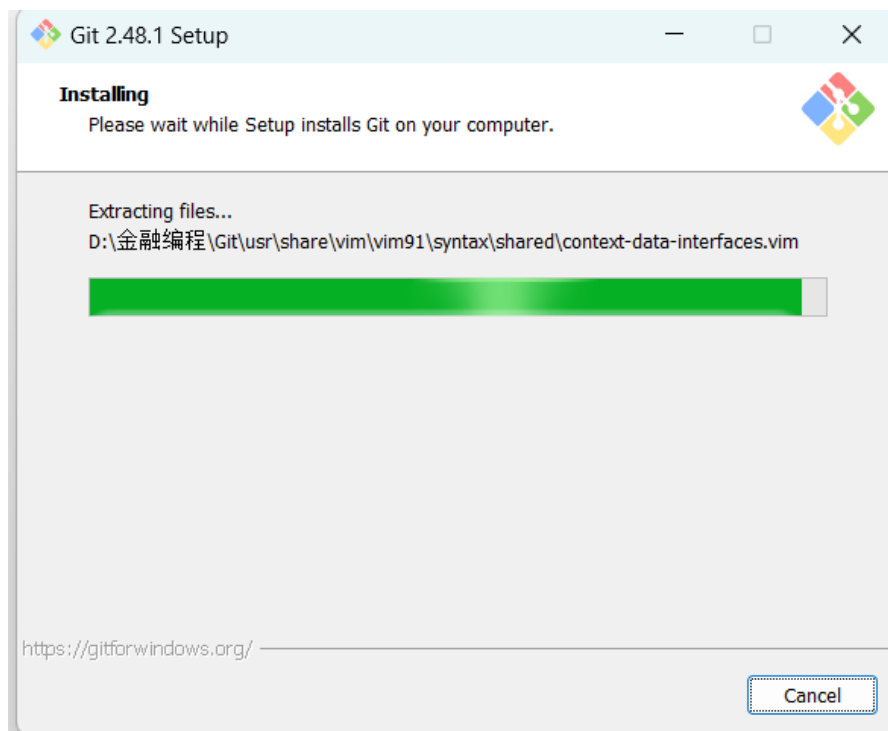
## 选择换行方式



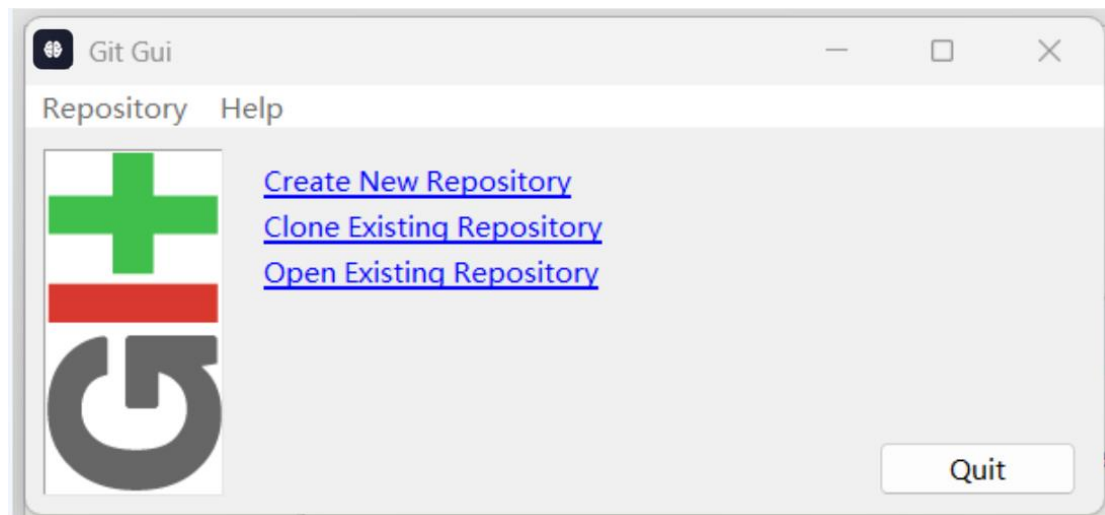
## 配置终端模拟器



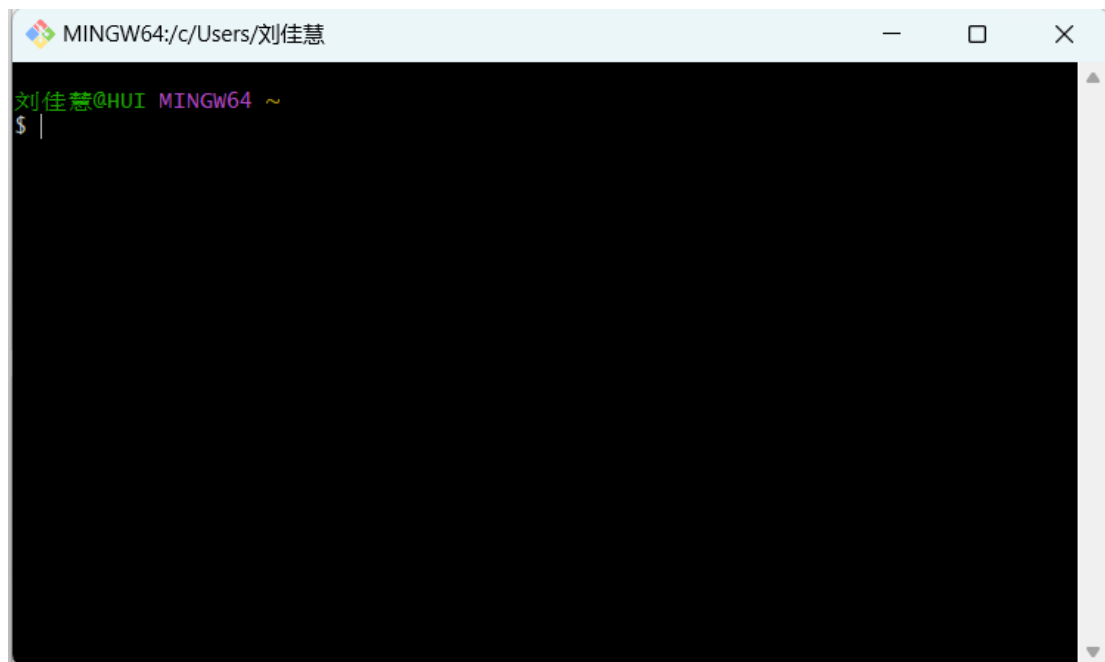
进行安装



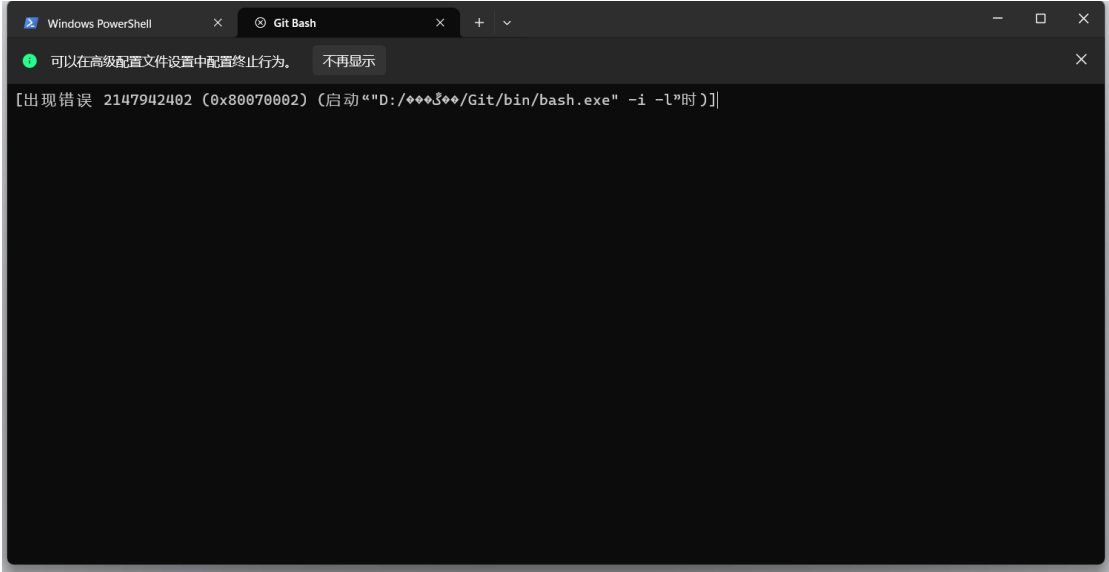
启动 Git 时遇到问题



启动时选择错误，已找到正确方式



在终端无法打开 git bash



尝试用豆包解决问题



在windows终端无法打开已经安装好的Git Bash 显示如下错误  
请帮我分析原因并提出解决方案

分享 ...

原因分析

- 1. **路径问题：**错误信息中 `D:/Git/bin/bash.exe` 存在乱码，这很可能导致系统无法正确识别 Git Bash 的可执行文件路径，从而无法启动。
- 2. **权限问题：**Windows 用户可能没有足够的权限来访问 Git 安装目录下的 `bash.exe` 文件，导致启动失败。
- 3. **软件冲突：**其他正在运行的程序可能与 Git Bash 的启动产生冲突，占用了相关资源。
- 4. **安装不完整：**Git 在安装过程中可能出现了错误，部分文件没有正确安装或损坏。

1. 检查路径：

- 打开 Git 的安装目录，确认 `bash.exe` 的实际路径。如果路径包含非英文字符，可能会导致问题，建议将 Git 安装在纯英文路径下。
- 在 Windows 终端中，手动输入正确的、无乱码的 `bash.exe` 路径来尝试启动。

2. 权限设置：

- 右键点击 Git 的安装目录，选择“属性”。
- 在“安全”选项卡中，确保当前登录用户拥有对该目录的完全控制权限。如果权限不足，点击“编辑”进行权限修改。

3. 关闭冲突程序：

- 打开任务管理器（快捷键 `Ctrl + Shift + Esc`），查看正在运行的程序。
- 关闭可能与 Git Bash 冲突的程序，如其他终端模拟器、占用端口的程序等，然后再次尝试启动 Git Bash。



可能的原因是安装路径中包含中文字符 修改文件名称



将 Git 安装的文件夹名改成英文之后，可按以下步骤操作：

1. **更新路径引用：**在 Windows 终端中，之前保存的 Git 路径因文件夹名更改而失效。需手动更新路径，在终端输入新的、正确的 Git 可执行文件路径，比如 `D:/newEnglishFolder/Git/bin/bash.exe -i -l` 来启动 Git Bash。
2. **检查环境变量：**
  - 按 `Win + R` 组合键，输入 `sysdm.cpl` 并回车，打开“系统属性”窗口。
  - 点击“高级”选项卡，选择“环境变量”。
  - 在“系统变量”或“用户变量”中找到与 Git 相关的路径变量（通常是 `Path` 变量），检查其中的 Git 路径是否已更新为新的英文路径，若没有，编辑变量值，修改为新路径。
3. **测试启动：**再次尝试在 Windows 终端中打开 Git Bash，看是否能正常启动，若仍有问题，可参考之前提到的权限设置、关闭冲突程序等方法进一步排查。

🔊 📄 ↗ 分享 ... | 🏠 🔒

在 Windows 终端中更新路径引用的详细操作如下：

1. 打开 Windows 终端，确保当前处于可以输入命令的界面。
2. 输入完整且正确的 Git Bash 可执行文件路径及启动参数：`D:\biancheng\Git\bin\bash.exe -i -l`。
3. 按下回车键，尝试启动 Git Bash。

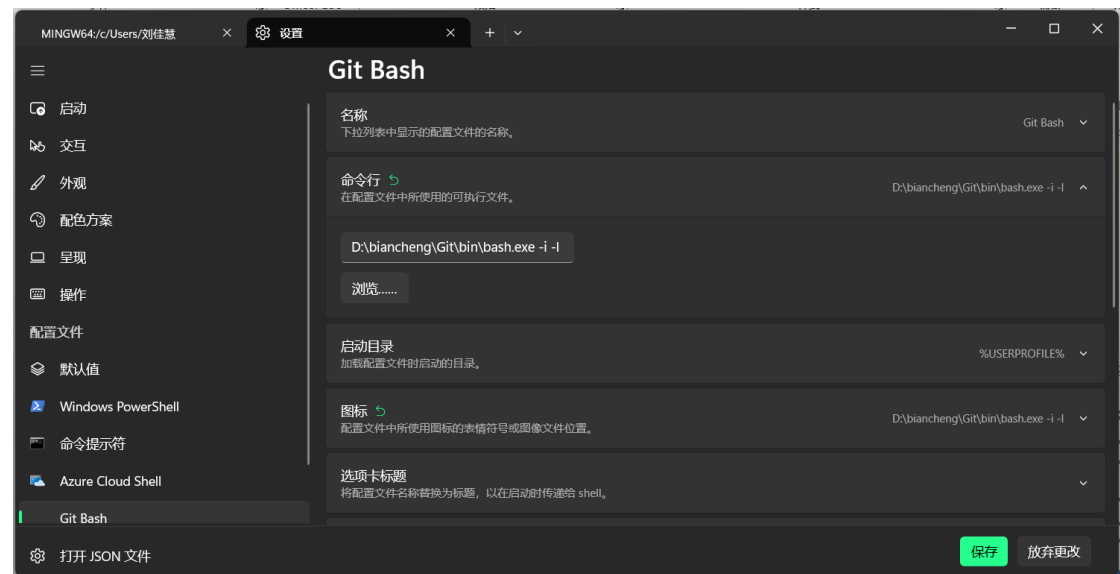
这样系统就能按照新的路径去寻找并启动 Git Bash 程序。

可以通过设置 Windows 终端配置文件来实现，步骤如下：

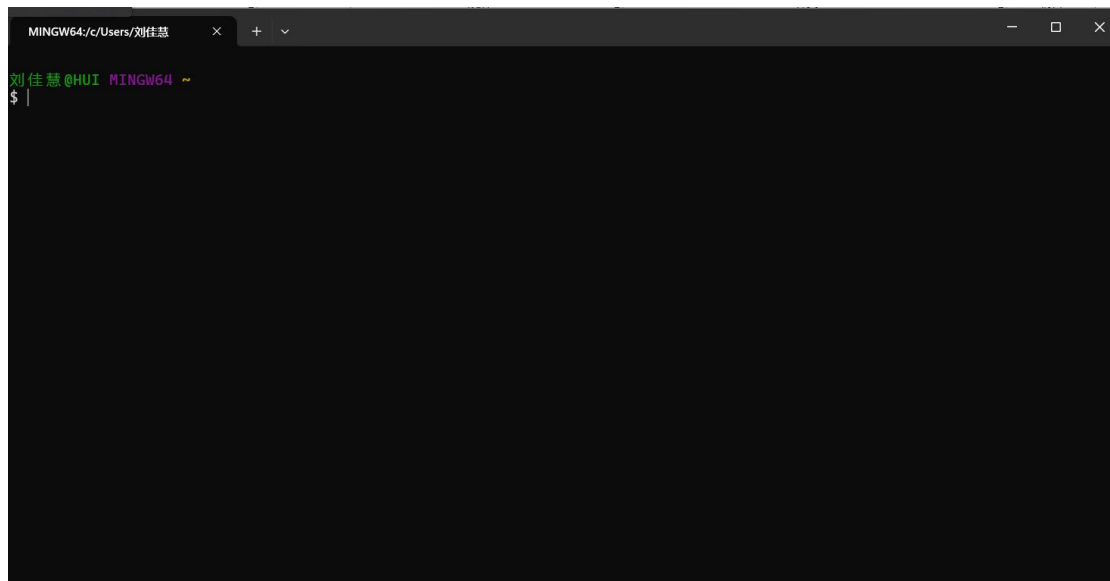
1. 打开 Windows 终端，点击窗口顶部的向下箭头图标，选择“设置”，或者按 `Ctrl + ,` 组合键直接打开设置页面。
2. 在设置页面左侧，找到并点击“配置文件”。
3. 在右侧找到 Git Bash 对应的配置文件，点击“编辑”。
4. 在打开的配置文件编辑页面中，找到“命令行”这一项，将其值修改为新的 Git Bash 可执行文件路径及启动参数，如 `D:\biancheng\Git\bin\bash.exe -i -l`。
5. 修改完成后，点击页面底部的“保存”按钮。

这样以后每次打开 Windows 终端并选择 Git Bash 配置文件时，就会按照设置好的路径自动启动，无需再手动输入路径。

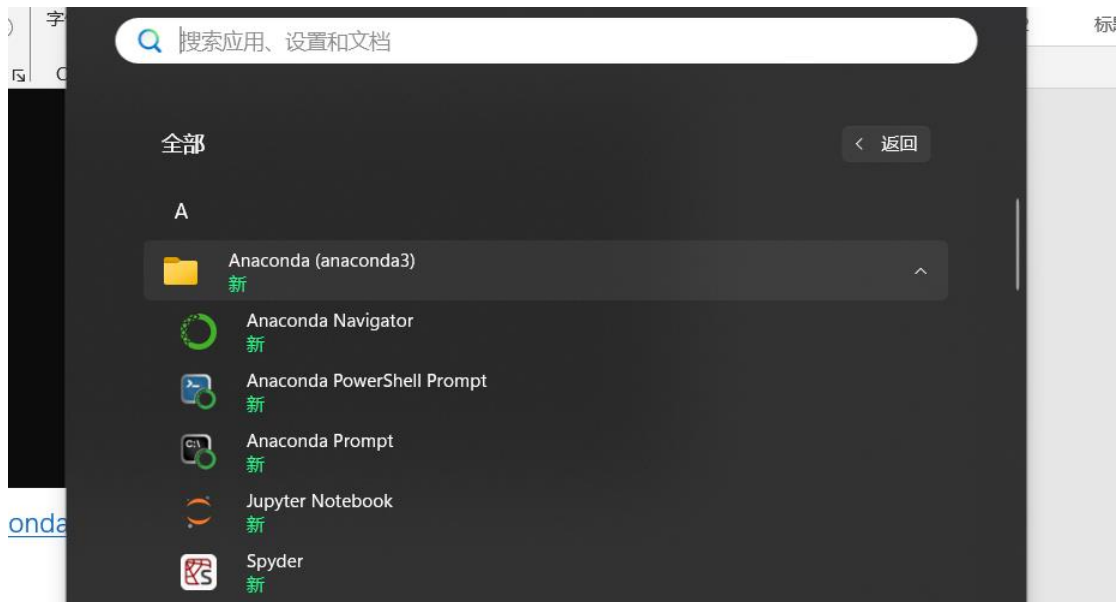
按照指示检查安装路径并修改为英文



成功解决问题！



3. 安装 [Anaconda](#) 安装完成!



```
MINGW64/c/Users/刘佳慧 x + v
刘佳慧@HUI MINGW64 ~
$ conda
usage: conda-script.py [-h] [-v] [--no-plugins] [-V] COMMAND ...

conda is a tool for managing and deploying applications, environments and packages.

options:
  -h, --help            Show this help message and exit.
  -v, --verbose          Can be used multiple times. Once for detailed output, twice for INFO logging, thrice for DEBUG
                        logging, four times for TRACE logging.
  --no-plugins           Disable all plugins that are not built into conda.
  -V, --version          Show the conda version number and exit.

commands:
  The following built-in and plugins subcommands are available.

COMMAND
activate      Activate a conda environment.
build         Build conda packages from a conda recipe.
clean         Remove unused packages and caches.
commands     List all available conda subcommands (including those from plugins). Generally only used by
             tab-completion.
compare       Compare packages between conda environments.
config        Modify configuration values in .condarc.
content-trust Signing and verification tools for Conda
convert       Convert pure Python packages to other platforms (a.k.a., subdirs).
create        Create a new conda environment from a list of specified packages.
deactivate    Deactivate the current active conda environment.
debug         Debug the build or test phases of conda recipes.
develop       Install a Python package in 'development mode'. Similar to 'pip install --editable'.
doctor        Display a health report for your environment.
export        Export a given environment
index         Update package index metadata files.
info          Display information about current conda install.
init          Initialize conda for shell interaction.
inspect       Tools for inspecting conda packages.
install       Install a list of packages into a specified conda environment.
list          List installed packages in a conda environment.
metapackage   Specialty tool for generating conda metapackage.
notices       Retrieve latest channel notifications.
```

## 4.创建密钥 SSH

### 遇到问题

```
MINGW64/c/Users/刘佳慧 x + v
刘佳慧@HUI MINGW64 ~
$ ssh-keygen -t ed25519 -C "2831601712@qq.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/刘佳慧/.ssh/id_ed25519):
Could not create directory '/c/Users/刘佳慧/.ssh' (No such file or directory).
Enter passphrase for "/c/Users/刘佳慧/.ssh/id_ed25519" (empty for no passphrase): |
```

### 用豆包分析解决问题

#### 方法一：手动创建目录

1. 打开文件资源管理器，导航到 `C:\Users\你的用户名` 目录（这里是你当前报错显示的路径部分，虽然乱码但能看出是 `Users` 目录下）。
2. 在该目录下手动创建一个名为 `.ssh` 的文件夹，确保文件夹权限设置为当前用户可读写。
3. 回到命令行窗口，重新运行 `ssh-keygen -t ed25519 -C "你的邮箱"` 命令，按照提示输入即可，此时应该不会再报无法创建目录的错误。

```
MINGW64:/c/Users/刘佳慧/.ssh  ×  +  v

刘佳慧@HUI MINGW64 ~
$ ssh-keygen -t ed25519 -C "2831601712@qq.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/刘佳慧/.ssh/id_ed25519):
Enter passphrase for "/c/Users/刘佳慧/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/刘佳慧/.ssh/id_ed25519
Your public key has been saved in /c/Users/刘佳慧/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:XyxyYuT0pdW8mzUzRircMWDkJIqPJmQvD0LebRiTruY 2831601712@qq.com
The key's randomart image is:
+--[ED25519 256]--+
|    o    . o    |
|  + . . . =    |
| .oE. . . +    |
|ooo.=o o . =    |
|..+o+o. S = B .|
| . *. + X + *   |
|   . + + + *    |
|       . = +    |
|       o        |
+-----[SHA256]-----+
```

## 新建 SSH 公钥

公钥名称 \*

2831601712@qq.com

公钥 \*

[了解如何生成 SSH Key.](#)

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAI0n0eB0rFln1lW5NklPhMzv2DRWu53GiUdmA/4/df 2831601712@qq.com

新建

取消

## 5.创建同名仓库



项目Fork中, 请稍后

我们正在为你fork原始项目, 请耐心等待。一般情况下, 复制过程很快 (个别大型项目可能需要更多时间)



```
刘佳慧@HUI MINGW64 ~/.ssh
$ cd

刘佳慧@HUI MINGW64 ~
$ cd repo

刘佳慧@HUI MINGW64 ~/repo
$ git clone git@gitcode.com:Aurora123456789123/week01.git
Cloning into 'week01'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (7/7), 8.75 KiB | 2.92 MiB/s, done.
```

## 完成克隆

```
① README.md ×
① README.md > # 提交 PDF 学习报告
1  # 提交 PDF 学习报告
2
3  ## 操作步骤
4
5  首先，将本课程仓库（你只能查看，不能修改）Fork 至你个人名下的同名仓库（你有修改权限），不必创建新分支。
6
7  然后，点击你个人仓库“代码”页面右上方的“Clone”按钮，复制你个人名下同名仓库的 SSH 地址（注意，不是`https://gitcode.com`开头的地址，而是`git@gitcode.com`开头的地址；前者只能下载，不能上推），在终端（Windows 可以用安装的 Git Bash）里，先用`cd`命令切换至你准备存放本地代码仓库的上级目录，然后用`git clone`命令把仓库克隆到本地。
8
9  如果能够在本地计算机文件夹里打开看到此文件的内容（推荐用 VS Code 打开你的本地仓库文件夹），就说明你成功地把本仓库 Clone 到本地了。
10
11 接下来，你可以把本周学习过程中记录产生的 PDF 报告复制或移动到仓库文件夹里来，然后在终端（Git Bash）里运行以下命令（需要适当指定参数）：
12
13  - 用`git add`命令将 PDF 报告加入版本管理
14  - 用`git commit`命令将代码改动（其实就是添加的 PDF 文件）确认提交到本地版本历史中
15  - 用`git push`命令将本地新提交的版本历史上推到 Gitcode 平台你个人名下的同名仓库中
16
17 命令操作成功的话，在 Gitcode 平台你个人名下的这个仓库里，就可以看到你自己推送上去的 PDF 报告了。
18
19 最后，在 Gitcode 上，从你的个人仓库发起 PR（Pull Request），目标为课程仓库（即 Fork 来源），才算完成了本周学习报告的提交。否则，就相当于你自己做了作业，但没有交给老师。
20
21 这套流程确实有点复杂。但选择走复杂的路，是为了实践获得更多的知识。这种“交作业”的方式，完全符合**现代开源协作开发实践**，有益无害。毕竟学习不是娱乐，不是享受，不是消费，而是为从事**生产**做好准备。
22
23 > 关于 Fork-PR 的知识可以参阅 [Gitcode 文档](https://docs.gitcode.com/docs/help/home/org_project/pullrequests/pr-fork)。对于每周的 PDF 报告，我们不进行代码审查与合并，你只要发起 PR 就可以了。
24
```