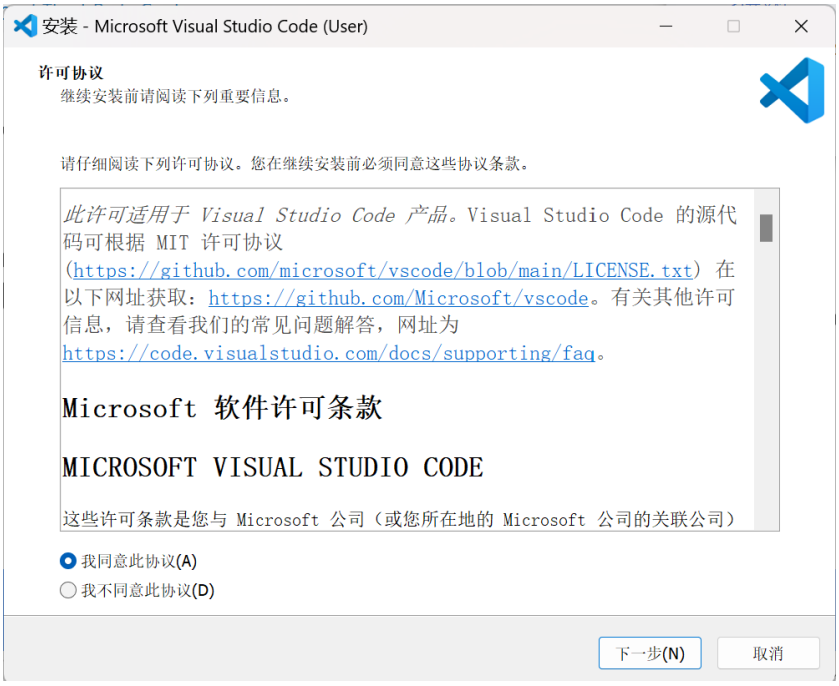
第一周 准备开发环境

任务一 安装 VS Code

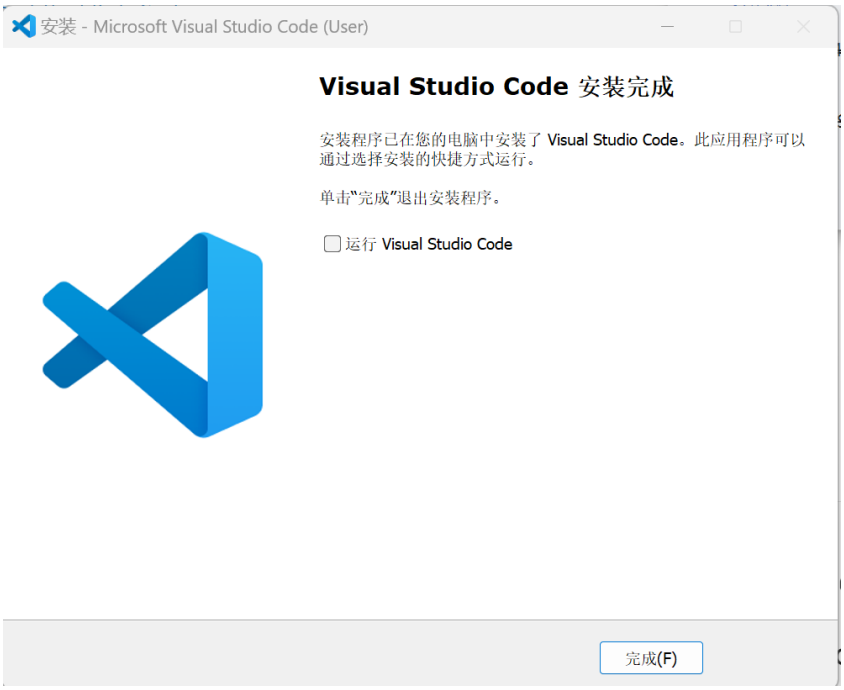
1.



2. 选择附加任务

- 勾选前两个选项方便在文件资源管理器中右键菜单栏打开 VS Code；
- 添加到 PATH：应用程序的启动路径

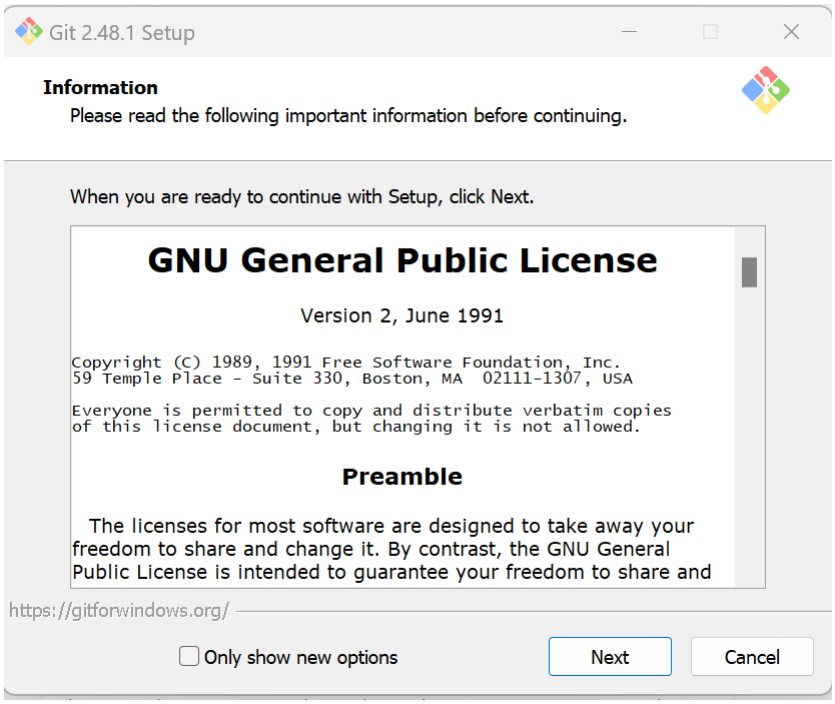
3. 安装完成



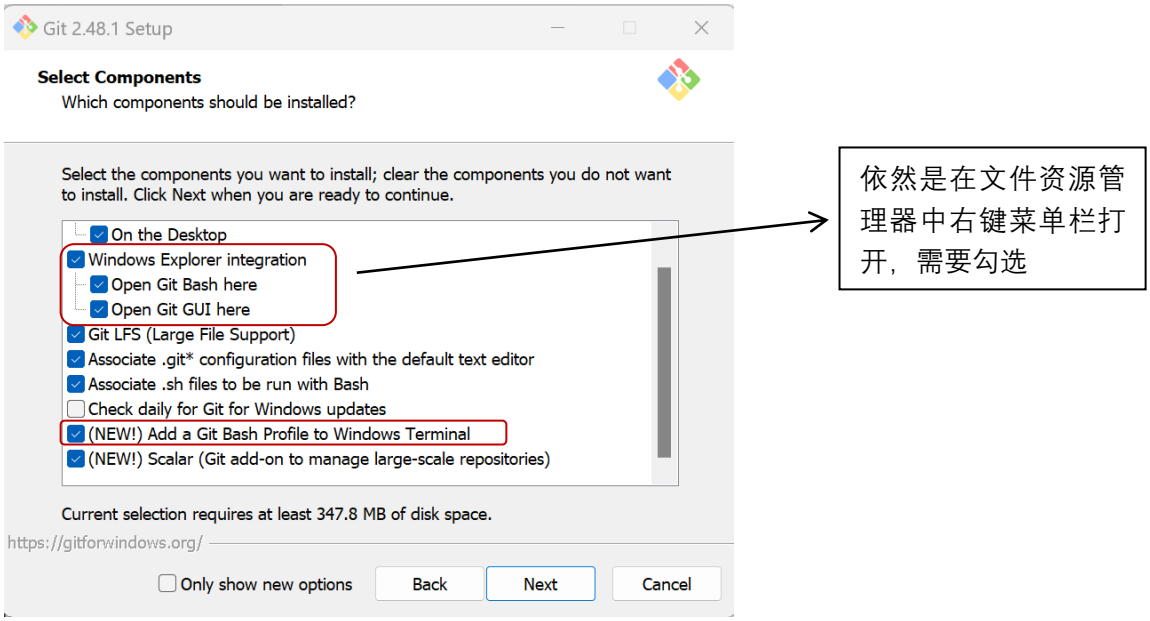
4. 固定到任务栏方便打开

任务二 安装 Git

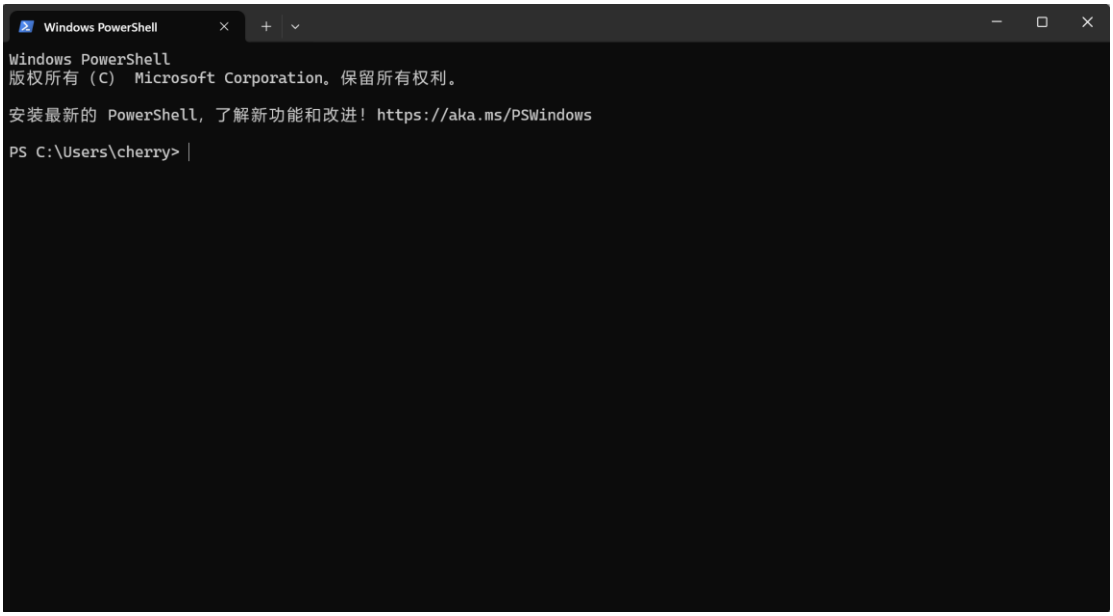
1.



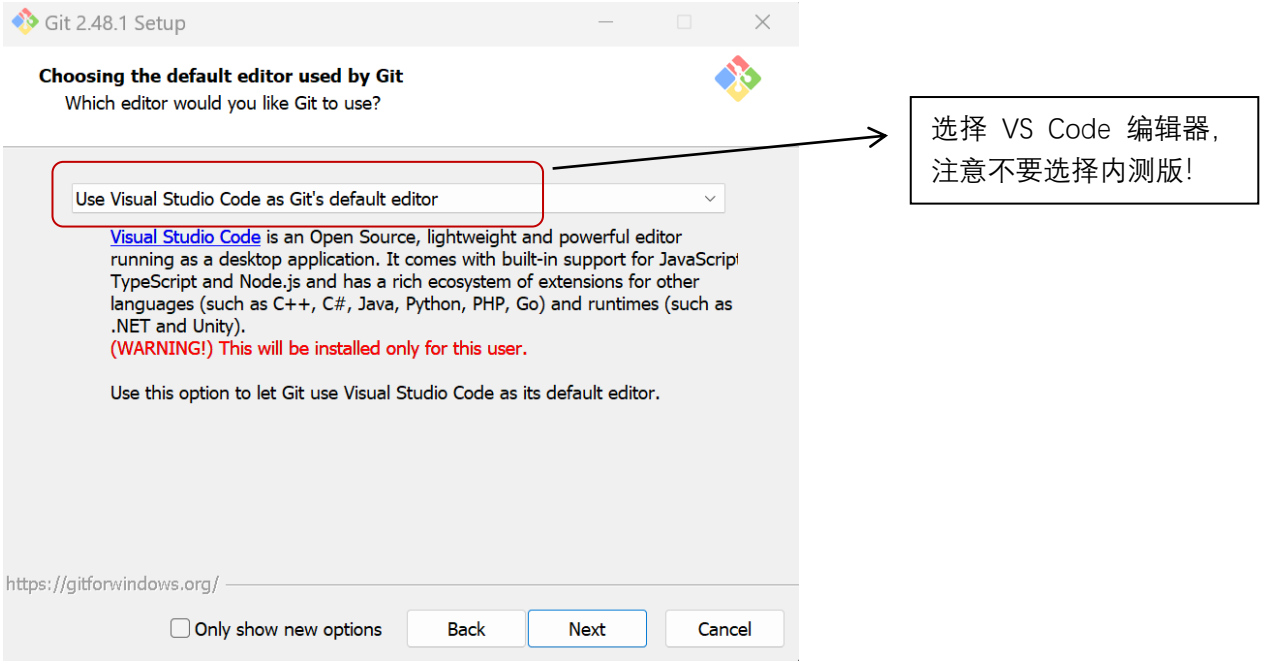
2.



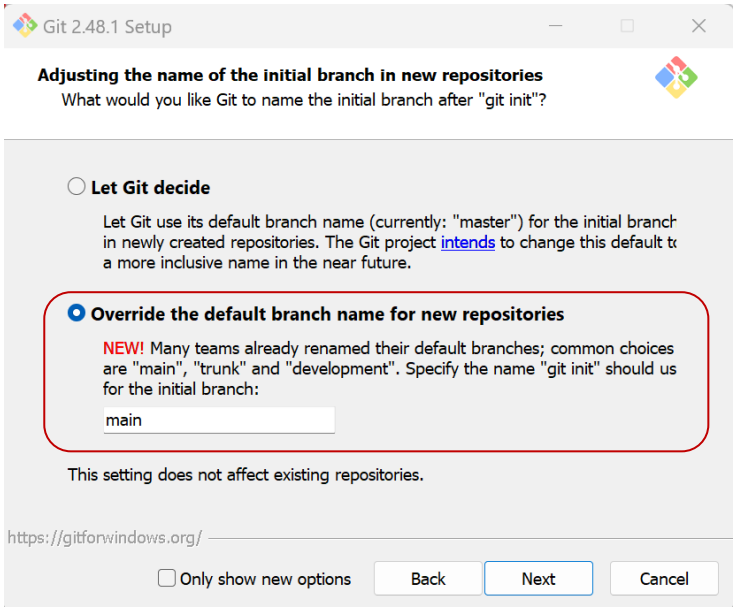
- Windows Terminal（终端）：可以将 Git Bash 配置加入到终端里，方便使用



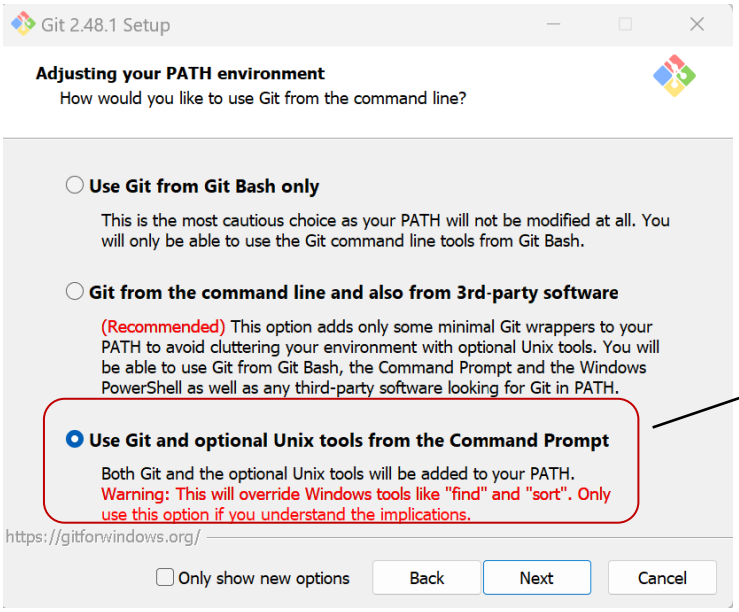
3. 选择编辑器



4. 选择 Git 分支

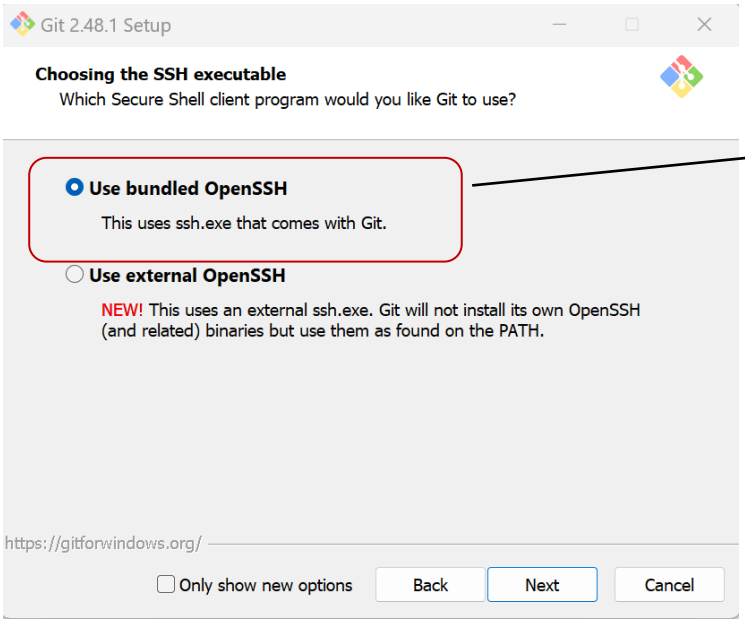


5.配置环境变量（PATH）：防止在终端中输入命令时查找不到



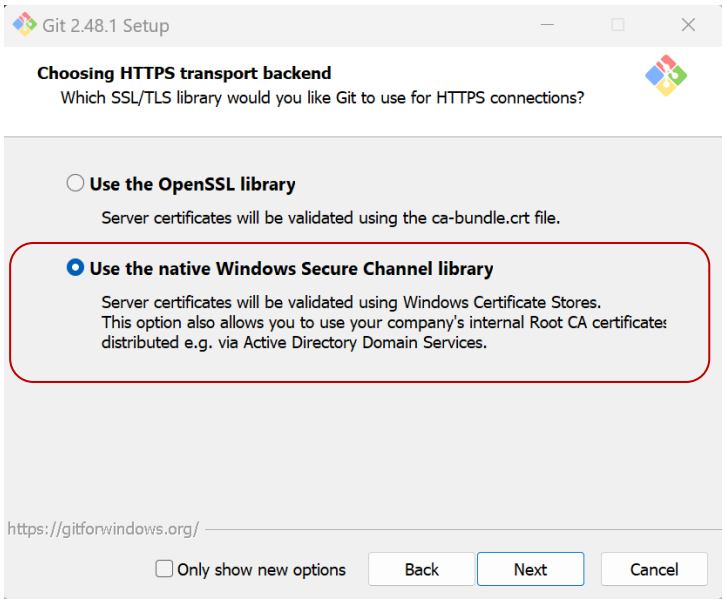
在命令提示符中不仅能够用 Git，而且能够使用其他的 Unix 工具

6.

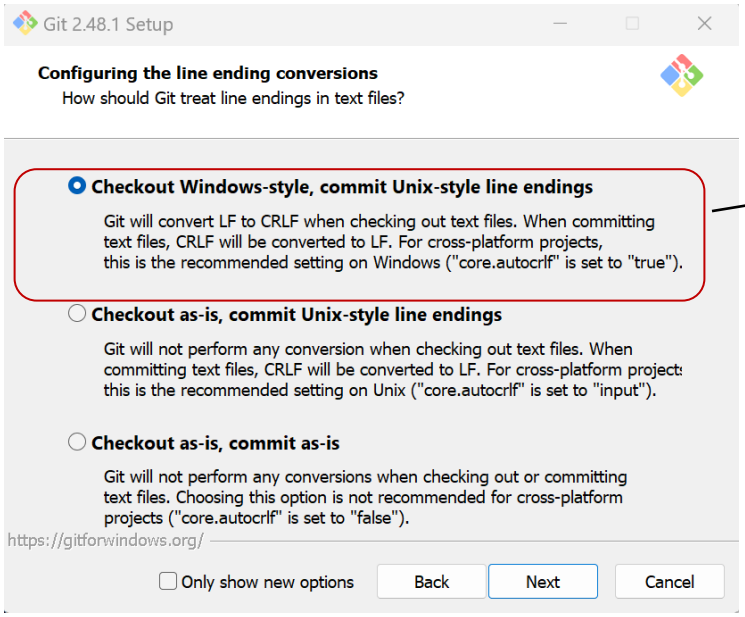


SSH：在互联网上加密通讯的工具

7.两个选项无明显差异

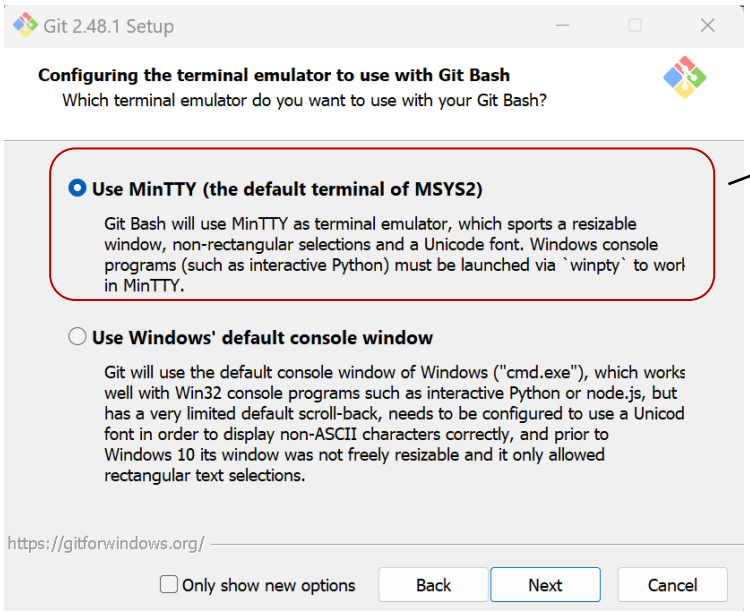


8．选择换行方式



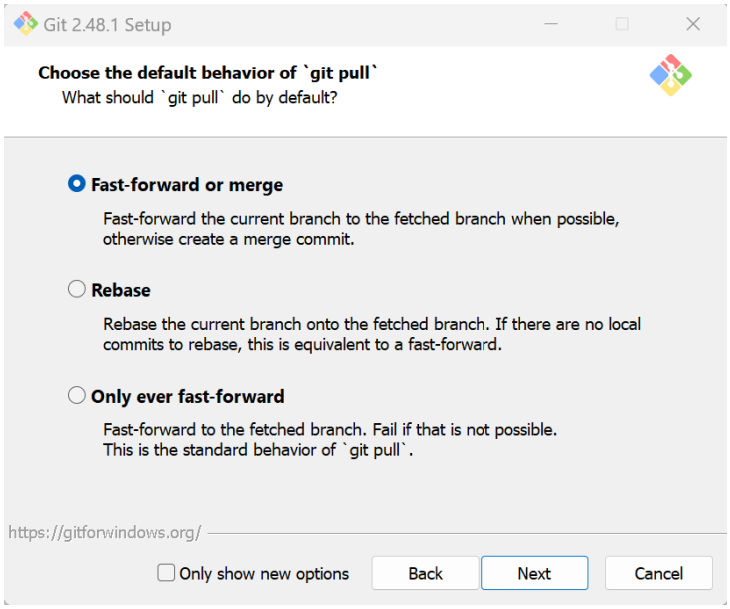
在代码迁出时是 Windows 风格，在向代码仓库提交代码时会自动转换为 unix 的换行

9. 配置终端模拟器

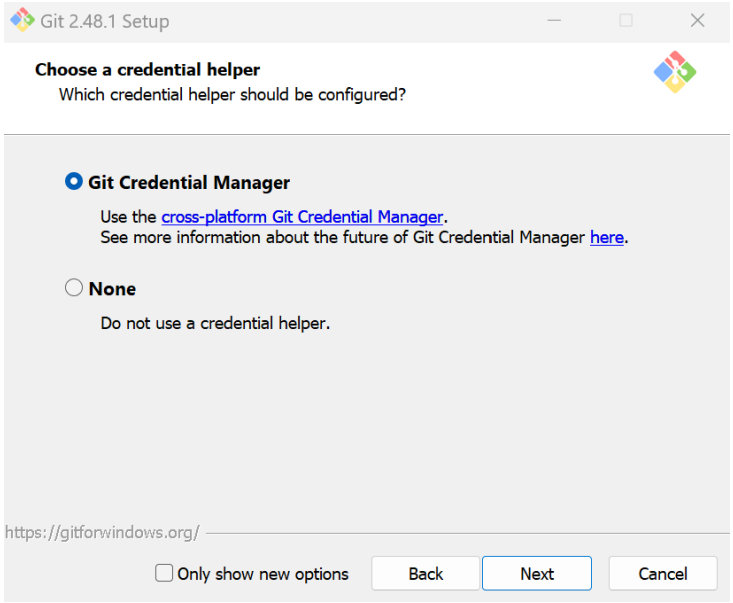


符合各种跨平台标准，虽然能在 Windows 上运行，但行为看起来和乌班图/苹果一致

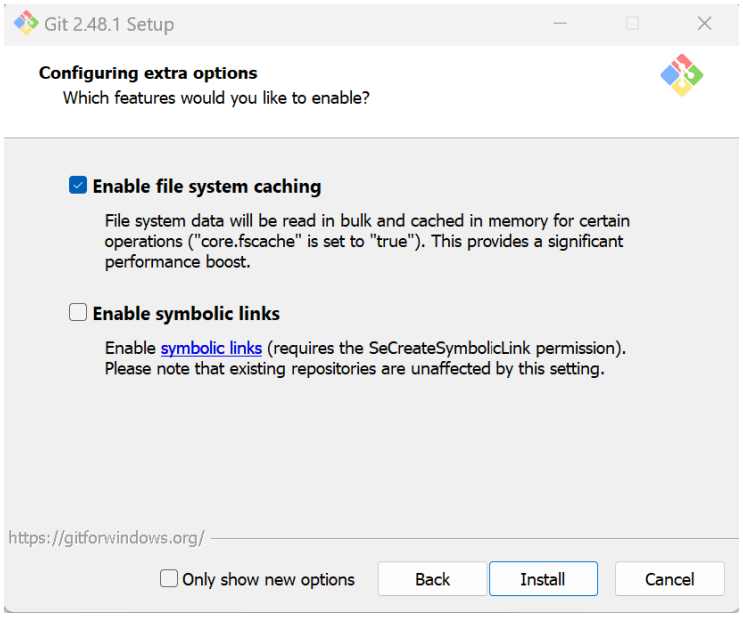
10. 和本地代码仓库保持同步的合并方式



11. 密码保存

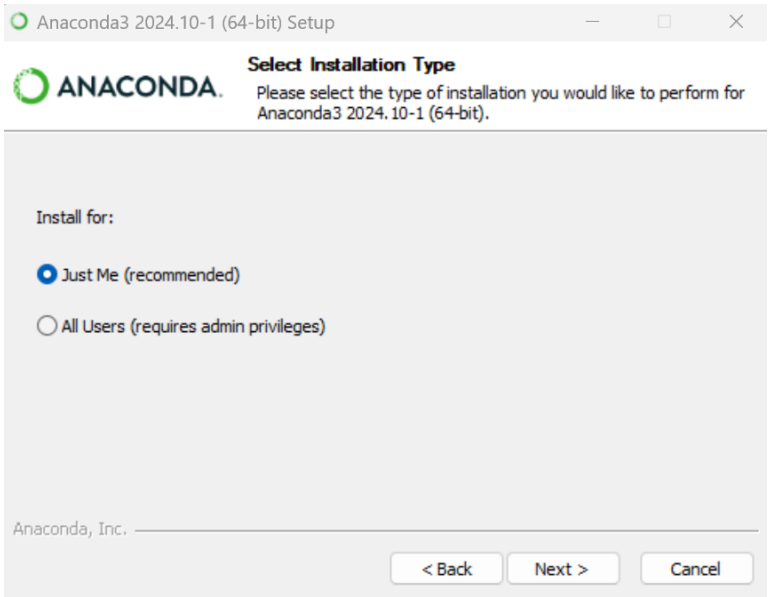


12. 文件系统的缓存

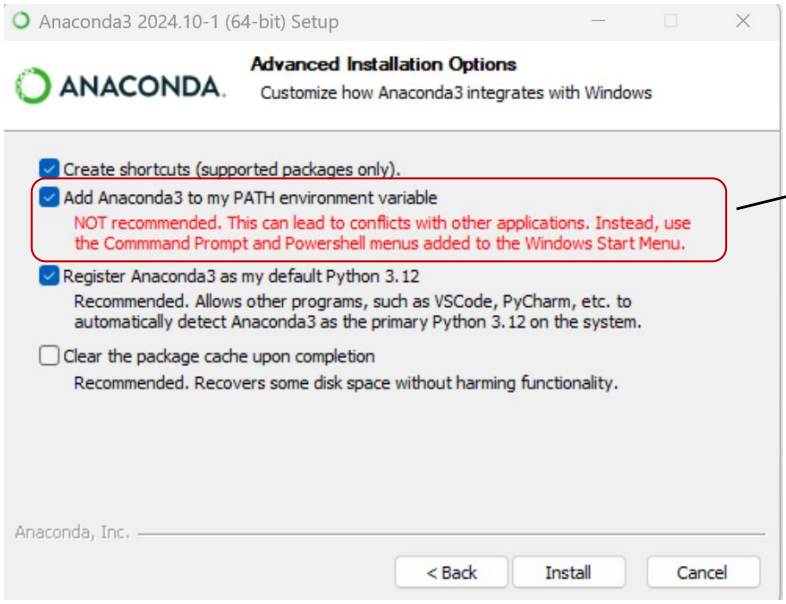


任务三 安装 Anaconda

1.

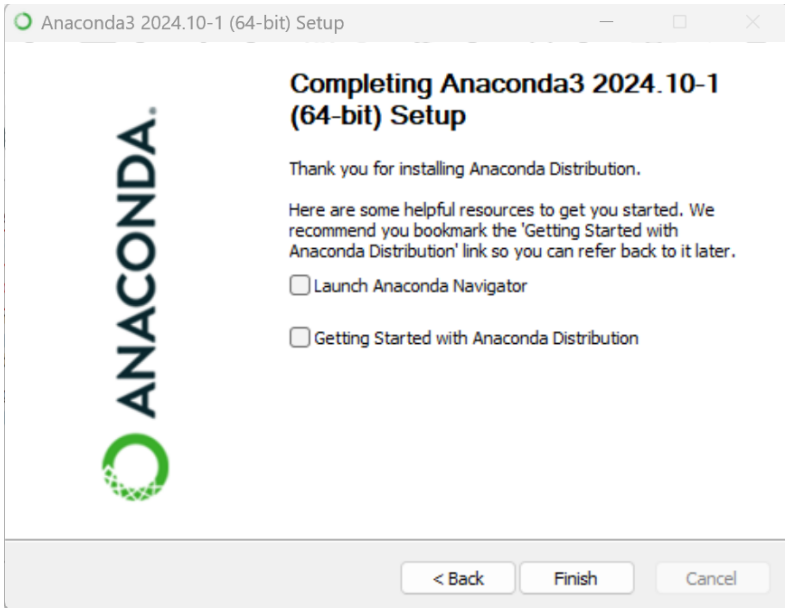


2.其他选项



添加到 PATH 环境变量，方便在终端中直接使用。

3.安装完成



4. 如何正确使用 Anaconda（在终端中使用）

```
MINGW64:/c/Users/cherry

    pack                See `conda pack --help`.
    package             Create low-level conda packages. (EXPERIMENTAL)
    remove (uninstall)  Remove a list of packages from a specified conda environment.
    rename              Rename an existing environment.
    render              Expand a conda recipe into a platform-specific recipe.
    repo                See `conda repo --help`.
    repoquery           Advanced search for repodata.
    run                 Run an executable in a conda environment.
    search              Search for packages and display associated information using the MatchSpec format.
    server              See `conda server --help`.
    skeleton            Generate boilerplate conda recipes.
    token               See `conda token --help`.
    update (upgrade)    Update conda packages to the latest compatible version.

cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ which conda
/d/Anaconda/Scripts/conda

cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ echo $PATH
/c/Users/cherry/bin:/mingw64/bin:/usr/local/bin:/usr/bin:/bin:/mingw64/bin:/usr/bin:/c/Users/cherry/bin:/c/Windows/syste
m32:/c/Windows:/c/Windows/System32/Wbem:/c/Windows/System32/WindowsPowerShell/v1.0:/c/Windows/System32/OpenSSH:/c/Progra
m Files (x86)/NVIDIA Corporation/PhysX/Common:/c/Program Files/NVIDIA Corporation/NVIDIA NvDLISR:/c/WINDOWS/system32:/c/
WINDOWS:/c/WINDOWS/System32/Wbem:/c/WINDOWS/System32/WindowsPowerShell/v1.0:/c/WINDOWS/System32/OpenSSH:/d/matlab/runtim
e/win64:/d/matlab/bin:/cmd:/mingw64/bin:/usr/bin:/d/Anaconda:/d/Anaconda/Library/mingw-w64/bin:/d/Anaconda/Library/usr/b
in:/d/Anaconda/Library/bin:/d/Anaconda/Scripts:/c/Users/cherry/AppData/Local/Microsoft/WindowsApps:/d/Microsoft VS Code/
bin:/usr/bin/vendor_perl:/usr/bin/core_perl

cherry@LAPTOP-QR2UKG4V MINGW64 ~
$
```

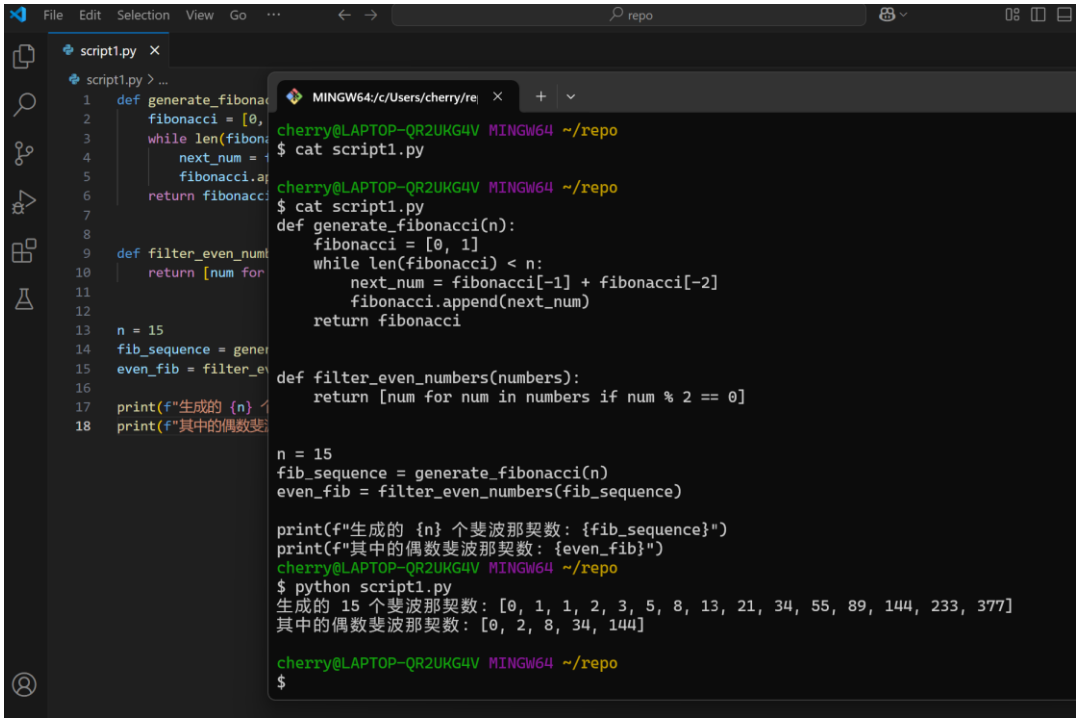
5. 使用 python

```
MINGW64:/c/Users/cherry
in:/d/Anaconda/Library/bin:/d/Anaconda/Scripts:/c/Users/cherry/AppData/Local/Microsoft/WindowsApps:/d/Microsoft VS Code/
bin:/usr/bin/vendor_perl:/usr/bin/core_perl

cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ python
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

6.新建一个目录 repo

```
cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ pwd
/c/Users/cherry

cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ mkdir repo
```

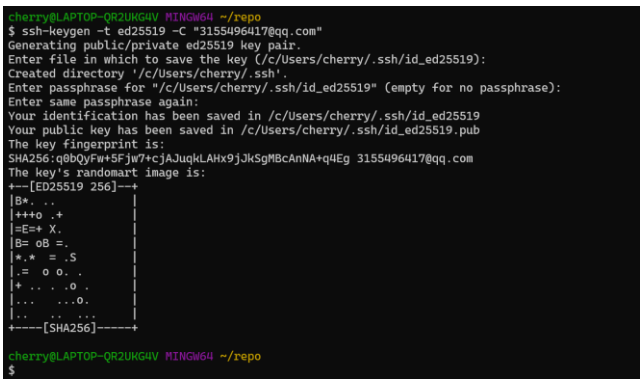
7.简单的运行一些 python 代码（生成斐波那契数列并进行筛选）

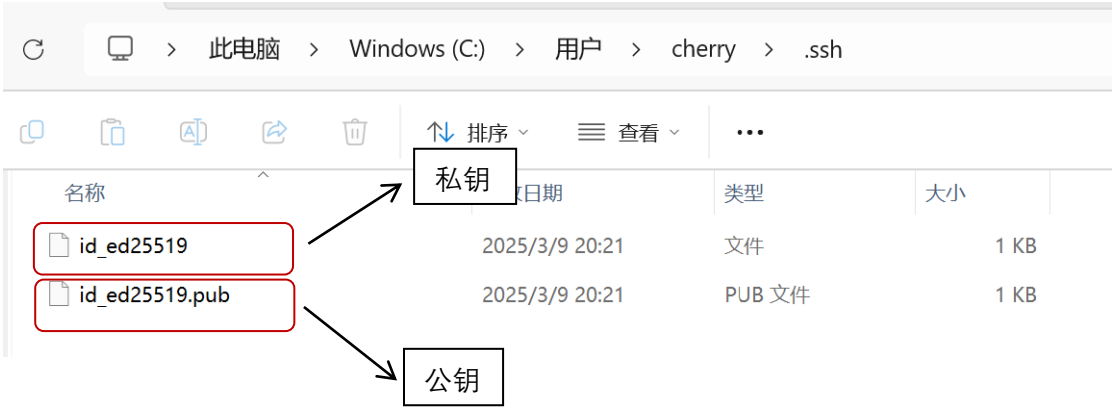


任务四 访问 GitCode (代码托管平台) 注册用户，然后保持登录

任务五 生成自己设备的 SSH 密钥 (公钥-私钥对)，将公钥添加到自己的 GitCode 安全设置里

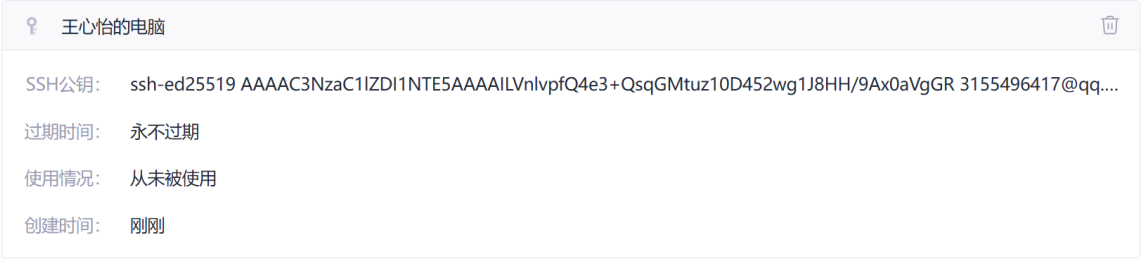
1.生成 SSH 密钥





- 他人用公钥加密发给我的内容我可用私钥解密查看（非对称加密，公钥可告诉他人，私钥不可）

2.添加到个人的 Gitcode 中



任务六 去 第 01 周打卡 仓库阅读说明，完成学习报告的提交

1. 把课程仓库 fork 至我个人名下
2. 把仓库克隆至本地

```
cherry@LAPTOP-QR2UKG4V MINGW64 ~/.ssh
$ cd

cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ pwd
/c/Users/cherry

cherry@LAPTOP-QR2UKG4V MINGW64 ~
$ cd repo

cherry@LAPTOP-QR2UKG4V MINGW64 ~/repo
$ pwd
/c/Users/cherry/repo

cherry@LAPTOP-QR2UKG4V MINGW64 ~/repo
$ git clone git@gitcode.com:Wxy496417/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.

cherry@LAPTOP-QR2UKG4V MINGW64 ~/repo
$
```