目录

[第1章 项目简介 1](#_Toc172072143)

[第2章 源代码和说明文档链接 1](#_Toc172072144)

[第3章 项目亮点 1](#_Toc172072145)

[1. 功能介绍 1](#_Toc172072146)

[2. 与其他版本控制软件的差异 1](#_Toc172072147)

[第4章 系统使用方式 2](#_Toc172072148)

[第5章 系统各模块分析 6](#_Toc172072149)

# 第1章 项目简介

本项目是一个类似于Git的产品，能够提供对同一类型的文件进行版本控制操作，比如文档，源码等文件。与Git这类版本控制软件不同的是，本项目是在本地创建一个文件夹来模拟Git中的本地仓库。

当用户每次提交一个文件时，会被视为提交了一个版本，然后提交的文件会按照“变量.时间戳”的命名方式来实现顺序保存在本地的文件夹中。同时，当用户提交一个版本时，会自动打开一个txt格式的日志文件，然后用户可以在里面写入所提交版本的备注信息（类似于git中的commit message），日志文件的命名格式会与所提交文件的命名格式一致，方便用户查看版本之间的关系。

# 第2章 源代码和说明文档链接

暂时先挂着。

# 第3章 项目亮点

### 1. 功能介绍

（1）可以记录每次提交的文件，命名格式按照变量+时间戳来实现顺序排列  
（2）每次提交时会打开一个txt格式的日志文件，用户可以在上面记录提交时要写的备注信息，类似于git中的commit message

（3）利用tkinter库创建出界面，用户可以在界面上看到对应的日志文件，然后也可查看日志内的信息，最后根据选中的日志可以点击“下载”按钮来下载该日志所对应的那份版本的文件

### 2. 与其他版本控制软件的差异

与Git相比，因为Git是以二进制文件来保存数据的（比如代码或者md格式文件），而word文档并非二进制，是一种带格式的文件。所以如果用Git就会导致每次都要完整的保存word文件，占用空间大且无法查看变化内容。（不过自己在电脑用的话空间问题不是大问题）

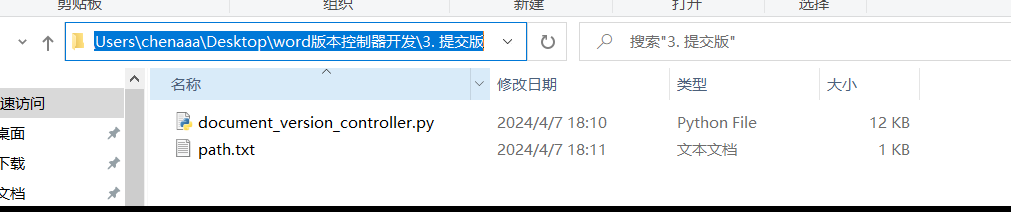
与其他版本控制软件相比，从网上查找的资料来看，主流的控制工具都是联网的，比如连接了onedrive这些网盘或者网站这种公共平台。而对于一些不想共享的私密性数据，通常情况下我们都是存在自己的本机中，但是这样会造成管理繁琐，比如自己经常会遗忘自己在哪个时间点改过哪些模块，又或者哪个是最新的文件。

所以本项目的思路就是做一个将内容存放在本机上（或者存放到另一台自己认为安全的服务器上）的版本控制软件，同时提供日志文件和记录提交时间的功能，这样用户就能根据日志信息找到自己想要的版本文件了。

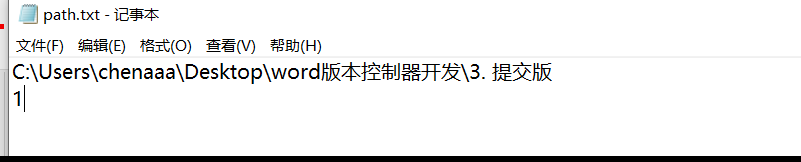
# 第4章 系统使用方式

最新更新：python代码文件已打包好为exe格式，修改后path.txt后直接运行exe文件即可（适用于没有python运行环境的用户）

1. 首先将两个文件path.txt和python代码文件放在同一文件夹下，然后记录下对应的文件夹绝对路径（注意文件夹下不能有document和log这两个文件夹，因为后面代码会自行创建）



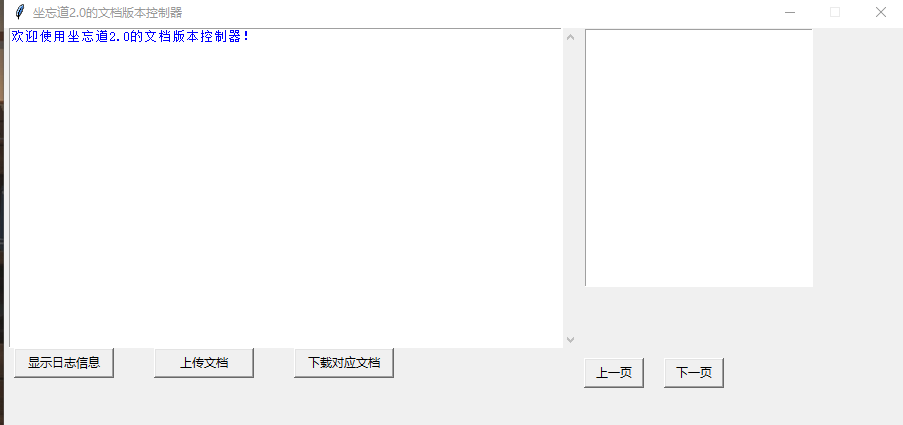
1. 打开path.txt文件，在第一行填入刚才记录下的文件夹绝对路径，第二行是版本起始号（通常填0或1都行）



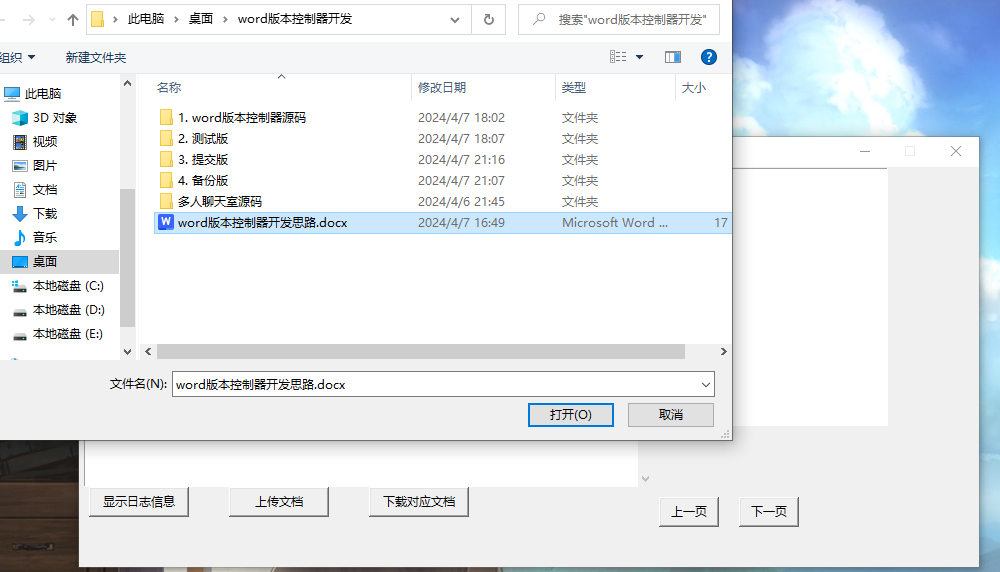
注意这个txt要为utf-8的格式（如下图所示）



1. 运行你的document\_version\_controller.py（或者exe文件）文件，成功后如下图所示



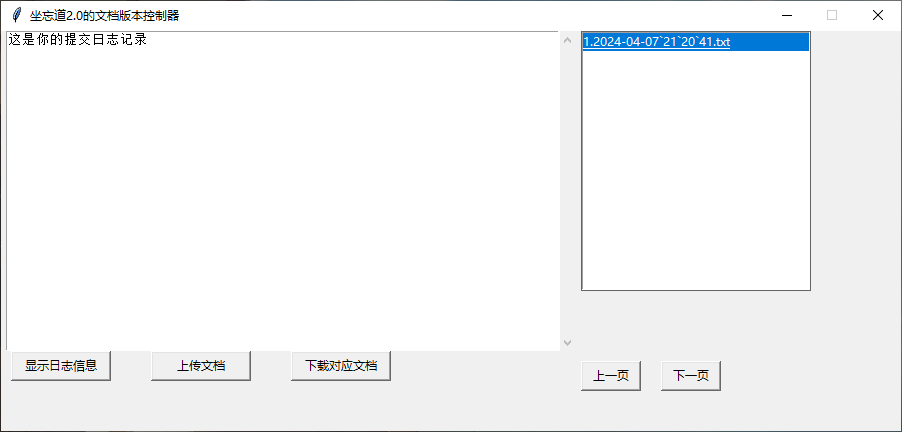
1. 点击“上传文档”，然后选择你要上传的文件（注意你上传的所有文件都要是同一类型的，第一次上传的文件类型决定了后面上传的文件类型，要换其他的需要从第1步重新开始），我这里就拿docx类型文件做示范



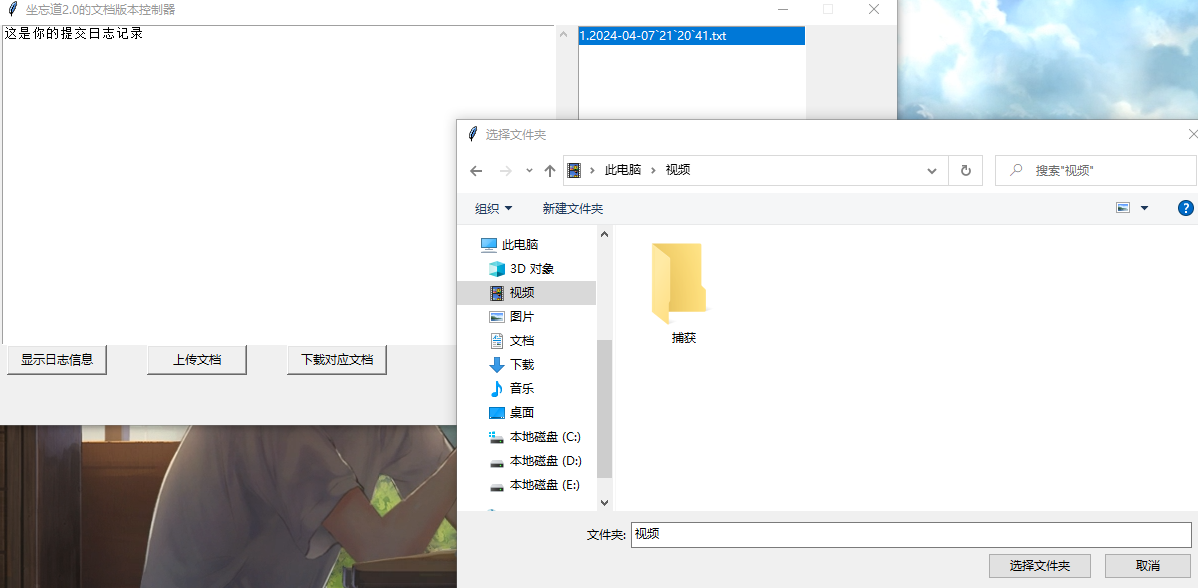
然后会弹出一个txt文件，你可以在里面写关于这个版本的备注信息，然后保存，关闭即可



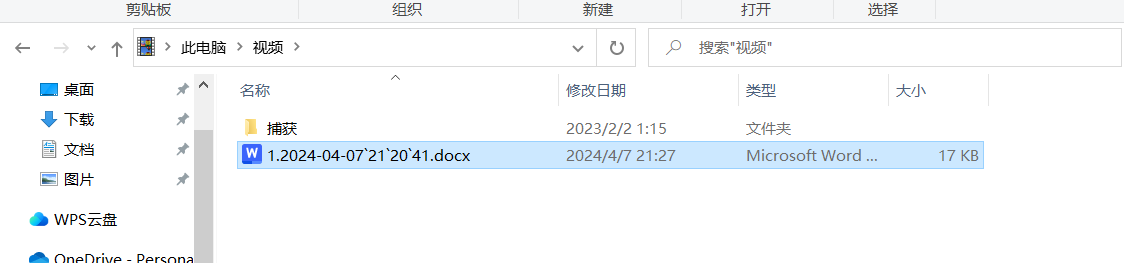
1. 此时右边的列表框就会出现对应的日志记录了，点击对应的日志，再点击“显示日志信息”即可在左边的显示框中看到日志文件中的信息



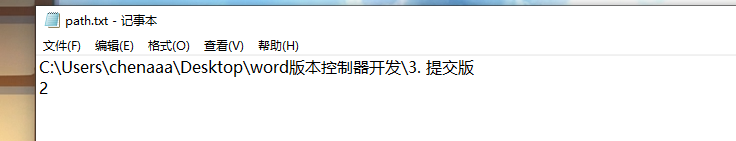
1. 点击对应的日志，然后点击“下载对应文档”，然后选择你要下载到的文件夹，即可下载到这个日志信息所对应的那份文档文件了



然后就能看到文档已经下载成功了

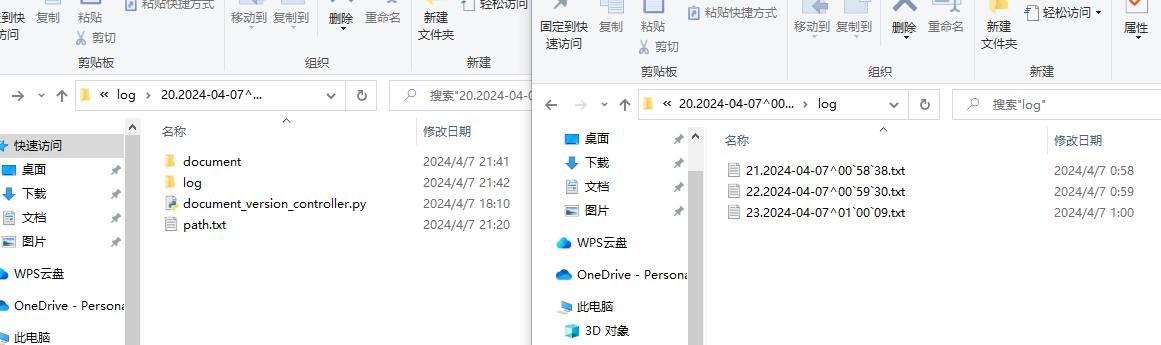
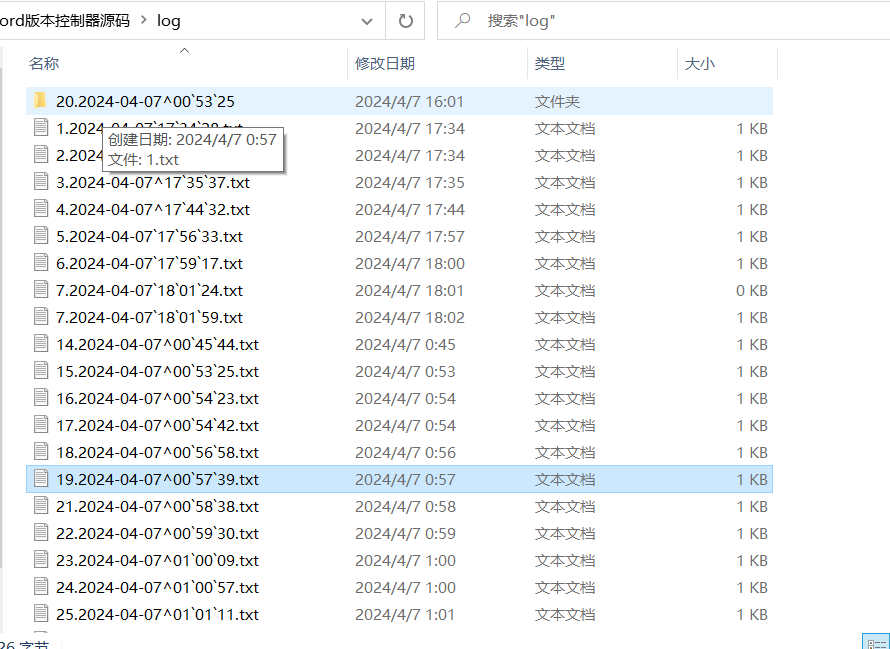


1. 此时再打开你的path.txt文件，可以看到第二行的版本号已经加1了



1. 如果你想回退到某个版本，并且在这个版本的基础上重新启动版本控制（Git中就是“换分支”这种操作），可以在log日志文件夹中按照对应的命名格式创建一个文件夹，然后把path.txt和python文件复制进去再次运行即可（path.txt中第二行的那个数字就是你的版本号数字，可以根据需要相应的修改。）

（比如我想要在20号版本文件的基础上开启一个新分支，那么我就创建一个新文件夹命名为20.xxxxxx，把path.txt和python文件复制进去，然后改path.txt中第二行数字为21,第一行路径也相应的更改，然后再运行python文件，那么下一个上传的文件版本就是21了，这样就可以简单的根据数字顺序看出你的版本号数了）



# 第5章 系统各模块分析

没空啦就先不写了，改天再写代码分析模块，没事可以自己下载源码来看