**command lin****e tool helper**

目录

[一、 概述 2](#_Toc185011248)

[二、 主要功能：（概括说明） 2](#_Toc185011249)

[三、 主要功能以及详细代码说明（详细说明）: 2](#_Toc185011250)

[1. 提供多种子命令： 2](#_Toc185011251)

[2. 命令生成及解释并提供给用户不同的选择： 3](#_Toc185011252)

[3. 提供不同选择： 6](#_Toc185011253)

[4. 与LLM聊天 7](#_Toc185011254)

[5. 通过配置文件自定义 API 密钥和接口地址： 8](#_Toc185011255)

[6. 记录命令历史，方便后续参考 9](#_Toc185011256)

[7. 记录日志 10](#_Toc185011257)

[8. 读取系统信息，以适配不同系统。 10](#_Toc185011258)

## 概述

comhelper 是一个命令行工具。通过与大型语言模型（LLMs，目前只能使用百度的文心一言）交互，该工具可以根据用户的需求生成、解释、执行和管理命令行命令，此外，该工具还支持配置管理、命令历史记录跟踪以及各种系统的适配。

## 主要功能：（概括说明）

1. **命令生成及解释：**通过与LLM的交互，根据用户的需求生成命令，同时生成命令的解释来让用户判断生成的命令是否正确。
2. **生成命令后提供给用户不同的选择：**编辑命令、执行命令、将命令复制到剪切板、退回到输入、退出程序
3. **通过配置文件自定义 API 密钥和接口地址：**默认的API为百度文心一言免费的接口，效果可能相对较差，用户可以根据自己的需求来更改付费API接口。
4. **记录命令历史，方便后续参考：**每次执行命令和将命令复制到剪切板时会将命令以及命令的解释和执行结果是否报错（执行命令时）保存，方便后续参考。
5. **与LLM聊天：**除生成命令外，用户也可以通过comhelper与LLM聊天。
6. **提供多种子命令：**用户可以根据不同的子命令选择不同的操作
7. **记录日志：**记录运行日志，为用户后续活动提供参考。
8. **读取系统信息，以适配不同系统。**

## 主要功能以及详细代码说明（详细说明）:

### 提供多种子命令：

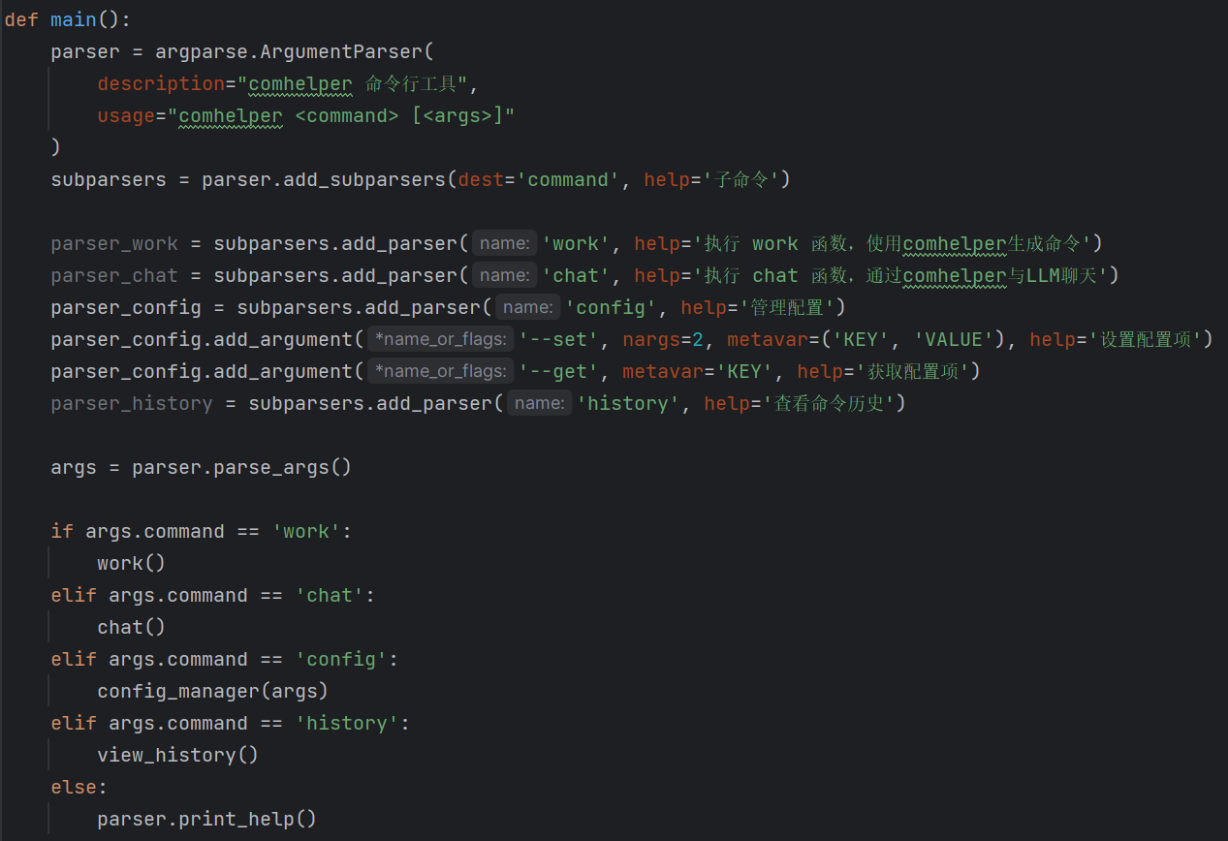
在环境中下载工具包后，用户可以通过comhelper命令来查看所有的子命令及功能：



由上述图片不难发现，comhelper有四个子命令，每个子命令分别负责不同的子功能，其中最主要的功能是“comhelper work”。

**代码解释：**

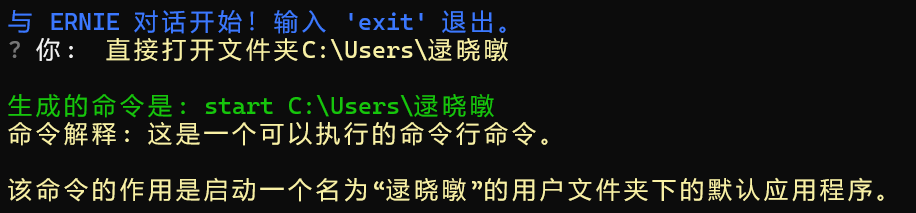
相应部分代码如下：



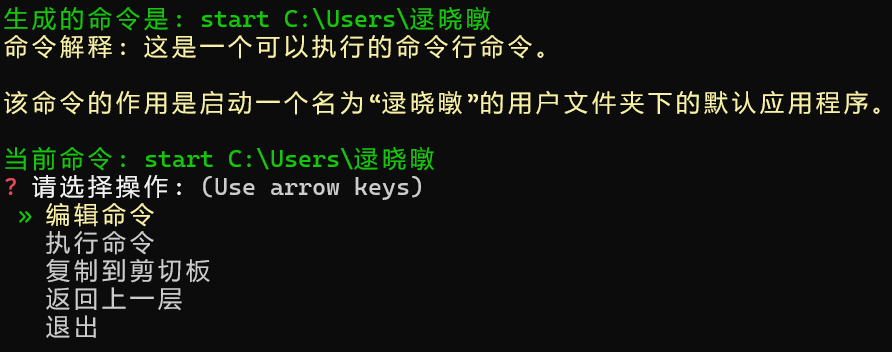
该部分代码使用 argparse 模块解析用户输入的命令和参数。并根据用户输入的命令，调用相应的功能函数。

### 命令生成及解释并提供给用户不同的选择：

用户运行“comhelper work”后就会唤醒对话，然后用户描述自己想要执行的操作，comhelper就会利用LLM生成对应的命令以及解释：

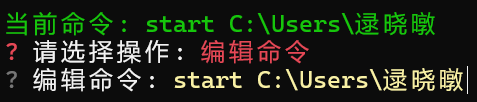


生成上述命令以及解释后，comhelper会给顾客提供几个不同的选择，用户可以通过键盘上下来选择：

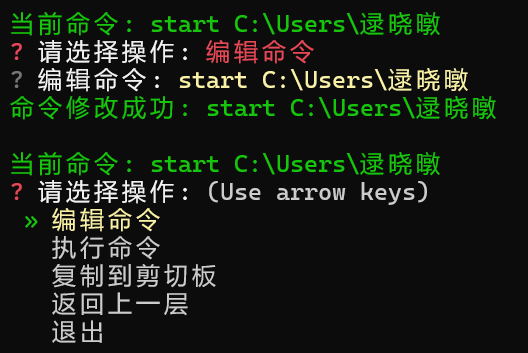


下面挨个说明不同选择的结果：

1. **编辑命令：**



选择编辑命令后，用户可以在当前的命令基础上修改命令，以达到用户想要的效果。编辑完成后，用户按下回车表示修改完成，会重新进行选择：



1. **执行命令：**

选择执行命令后，会显示警告信息，并让用户选择是否执行：

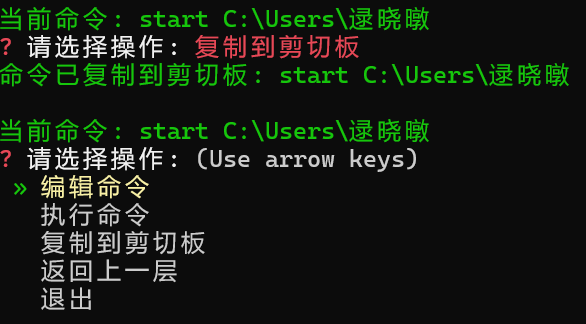


用户确定执行后，会执行命令，并输出执行结果，如果执行失败也会输出错误信息。



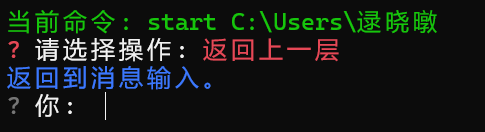
1. **复制到剪切板：**

选择复制到剪切板后，comhelper会将生成的命令复制到剪切板并重新进行选择：



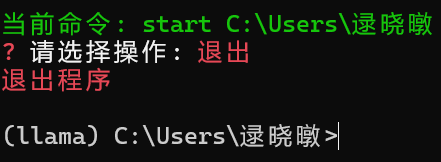
1. **返回上一层：**

选择返回上一层后，会返回到消息输入:



1. **退出：**

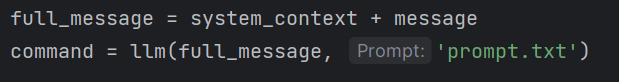
选择退出后，会退出程序：

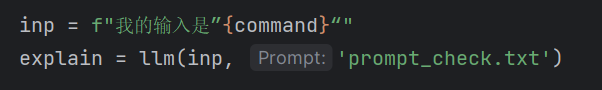


**解释代码：**

（这部分代码比较复杂，所以只解释比较关键的部分，完整代码在附件中）

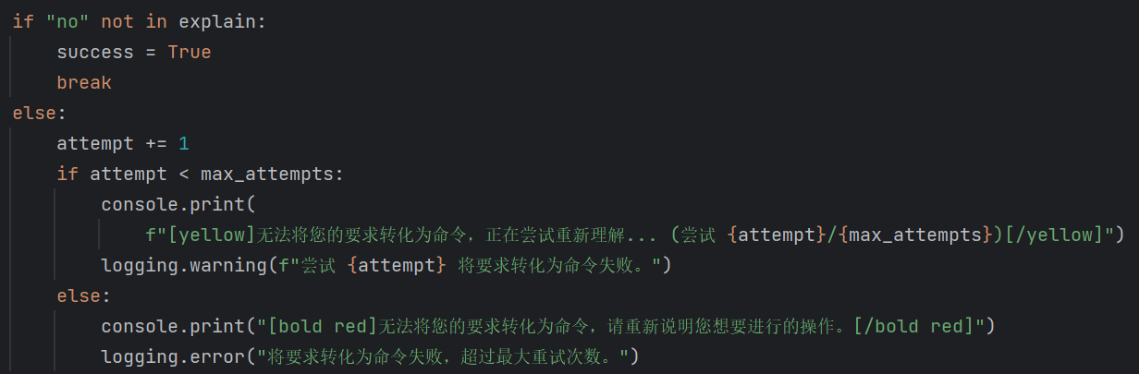
1. **一次对话两次使用llm：**



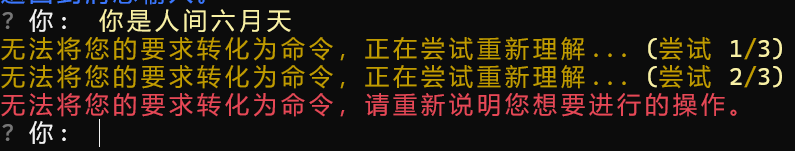


如上图，在一次对话中，会调用两次llm，第一次生成命令，第二次则对命令进行验证，如果生成的命令可以执行，则输出命令的解释，如果生成的不是可以执行的命令，则返回no（no的作用后面说明）。其中system是读取的系统信息，以保证llm生成的命令能够适配系统。

如果返回no，则会进行重试：



重试次数达到3次时，则会让用户重新说明，效果如下：



这样的两次llm调用和3次重试，可以极大提升生成命令的准确率，防止生成的命令不可以执行，影响用户体验。

### 提供不同选择：

提供不同选择对应的代码如下：



使用questionary.select 提供操作选项，并且可以通过键盘控制光标来实现选择。选择某一选项后，可以通过不同action值来执行不同的操作。

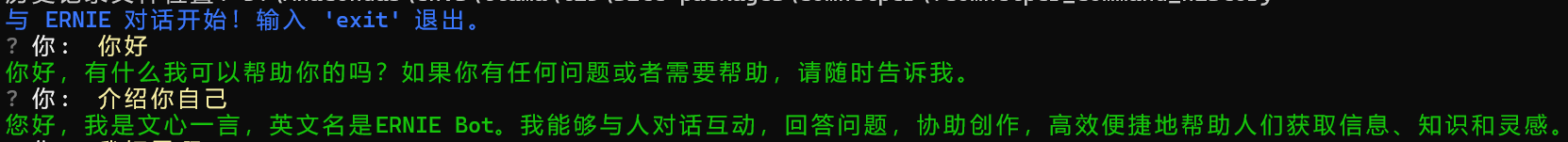
1. **命令执行：**



命令执行部分使用单独定义了一个函数，使用subrprocess.run来执行命令。

### 与LLM聊天

用户调用“comhelper chat”后，同样会开启一个对话，但不同于之前的生成命令，这个对话就是单纯的与llm聊天：



这个功能的实现要简单的多，只需要将llm的回复返回到命令行中即可，所以在这里简单说明llm调用的函数，事实上“comhelper work”也会调用这个函数。

**解释代码：**



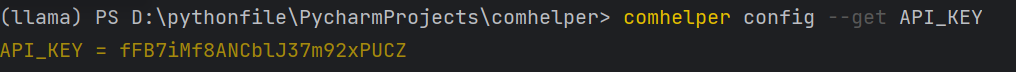


上述代码即为调用文心一言API的代码的主要部分。其中中的llm\_base\_url 会从配置文件中读取，access\_token的获取也会用到从配置文件中读取的API\_KEY和SECRET\_KEY。这样可以通过改变配置文件来改变调用的文心一言的模型。

### 通过配置文件自定义 API 密钥和接口地址：

默认的API为百度文心一言免费的接口，效果可能相对较差，且密钥为我本人的密钥，因为安全问题，在一段时间后我可能会关闭这个模型的密钥。

通过“comhelper config --get key”来查看配置文件中的值：



通过“comhelper config --set key value”来修改配置文件中的密钥和接口地址：



解释代码：

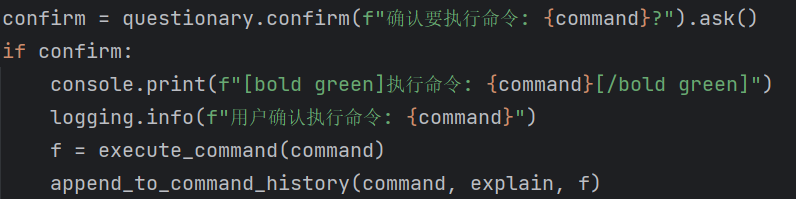




上述代码为管理配置文件的代码，针对解析出的不同值来进行不同的操作。

### 记录命令历史，方便后续参考

在每次执行命令和将命令复制到剪切板时会将命令以及命令的解释和执行结果是否报错（执行命令时）保存，相应代码如下：

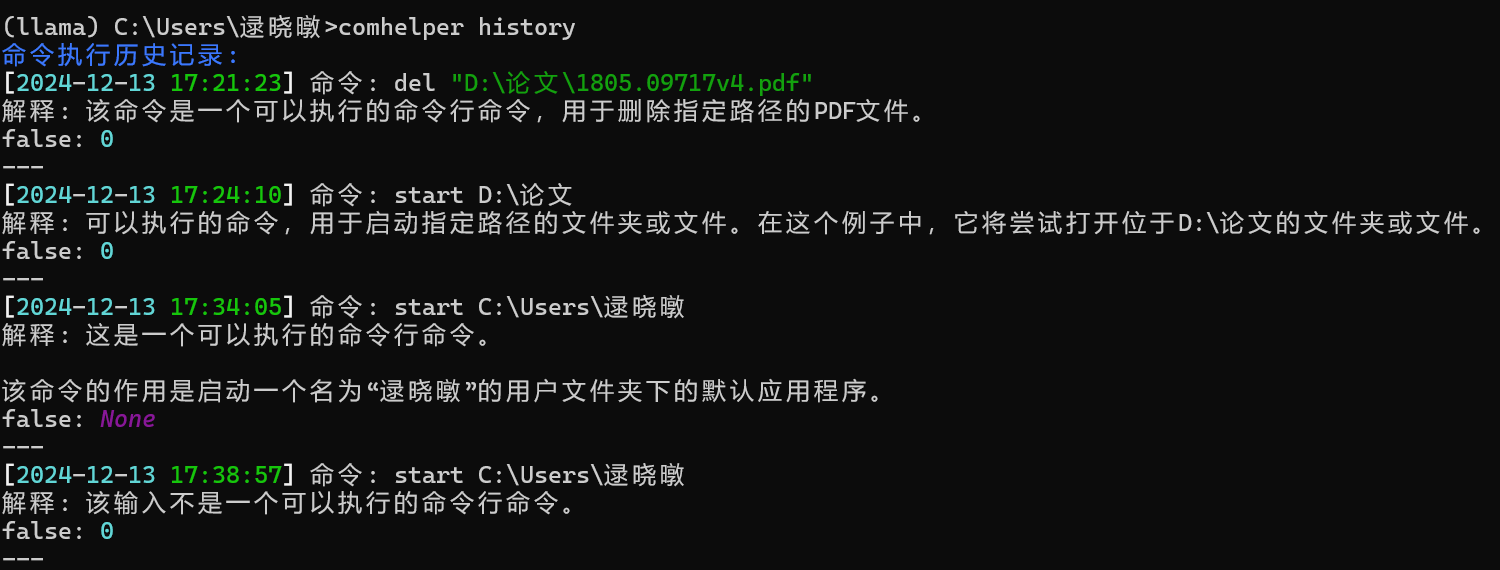


在用户确定执行目前命令时会将当前命令以及相关信息保存，在用户选择复制到剪切板时也会进行类似操作。其中append\_to\_command\_history的代码如下：



在调用该函数时，会将命令以及相关信息保存在文件中。

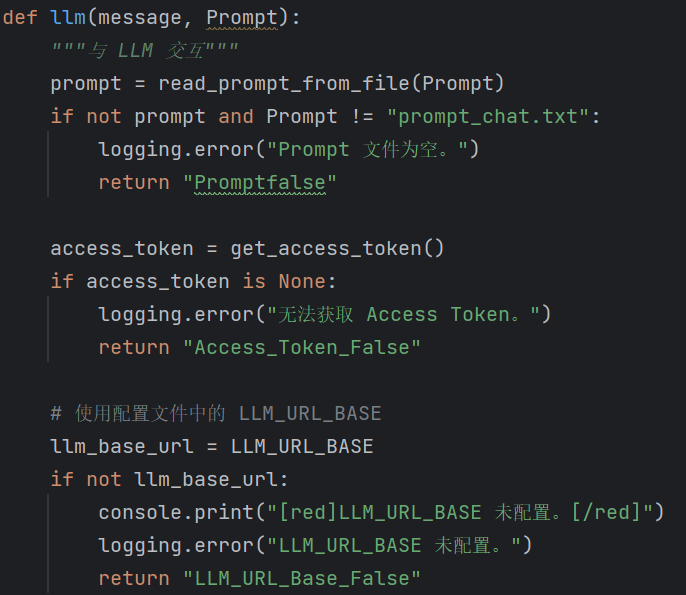
运行“comhelper history”时，会输出保存的命令以及相关信息：



由于这个功能的代码就是简单的读文件，所以这里不再粘贴相关代码了。

### 记录日志

comhelper会在运行的各个节点记录日志，记录一些相关信息：



如图，几乎每个函数都会包含多条日志记录信息。

### 读取系统信息，以适配不同系统。

comhelper会自动读取系统的相关信息，并在与llm交互时将系统信息与用户输入一同传递给llm，以保证llm生成的命令与系统相适配。

下述是读取系统信息的代码：

