**测试结果**

**注：**由于该命令行工具使用python编写，对python的软件包有依赖，首先需使用pip进行安装，且在使用pip安装时会自动安装一些依赖的软件包，为了防止与原本的电脑环境产生冲突，所以建议在虚拟环境或docker中安装此包。

**安装此包步骤：**

1. 下载代码压缩包，并将下载的压缩包解压；
2. 在命令行中打开虚拟环境；
3. 在命令行中打开减压后的文件夹，确保当前工作目录与setup.py所在目录相同；
4. 执行命令：pip install .
5. 执行成功后即已经成功安装此包，之后在该虚拟环境中即可使用comhelper。

**下面在两种系统中对该comhelper进行测试：**

1. **Windows：**
2. **运行comhelper：**

运行comhelper后会输出简单介绍和所有子命令：



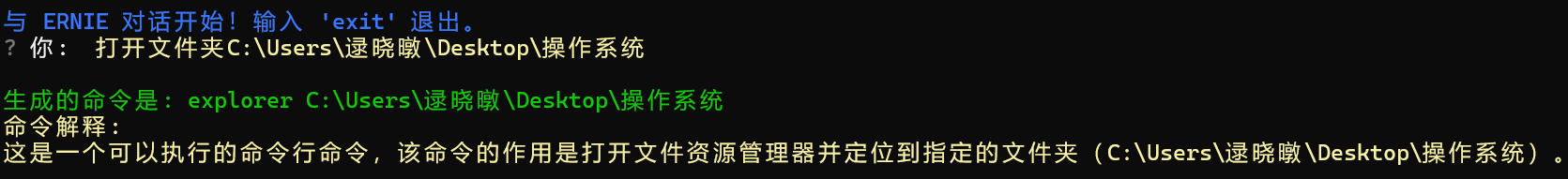
1. **运行comhelper work：**

运行comhelper work后会激活对话，用于根据输入生成命令：

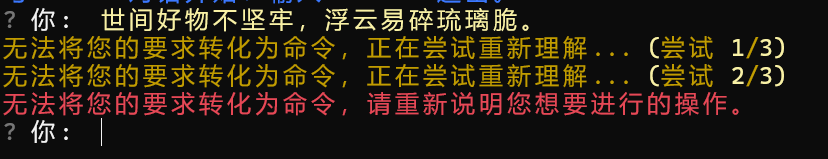


在最开始会输出目前的系统信息，供用户检查，防止生成不适配系统的命令。

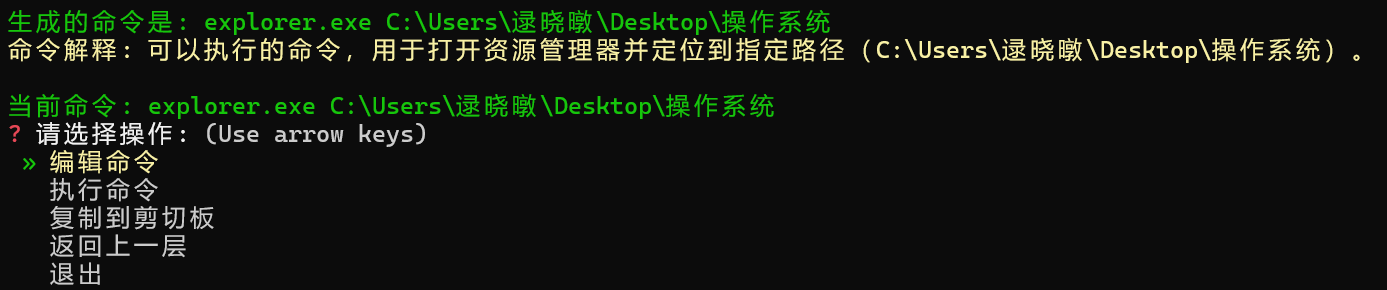
用户输入要求后，会生成对应的命令以及解释：



而当用户输入的要求并不是一个可以转换的要求时，会进行三次重新尝试，如果确定不是一个可以转换的要求就会让用户重新说明要求：

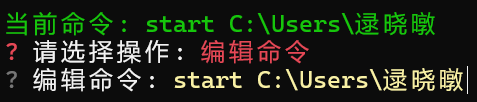


用户输入成功转换成命令后，会进入选择状态，用户通过键盘选择几种操作：

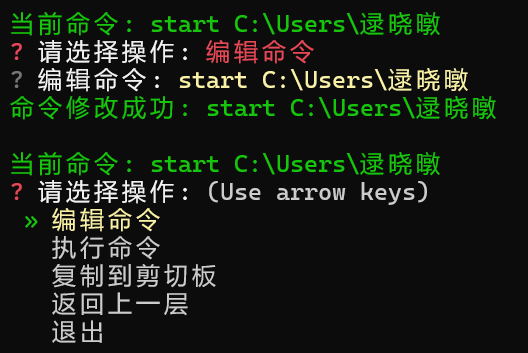


下面挨个说明不同选择的结果：

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：

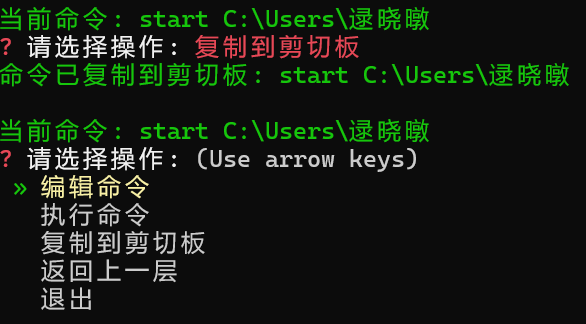


用户确定执行后，会执行命令，并输出执行结果，如果执行失败也会输出错误信息。



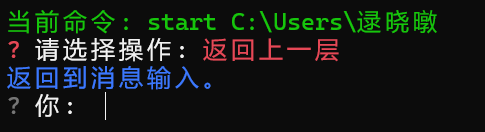
1. **复制到剪切板：**

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：



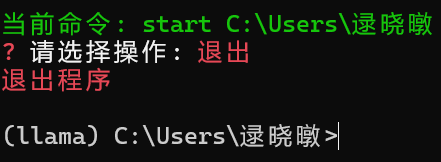
1. **返回上一层：**

选择返回上一层后，会返回到消息输入:



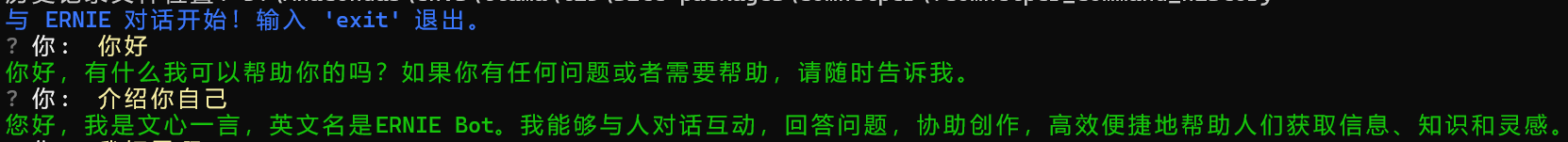
1. **退出：**

选择退出后，会退出程序：



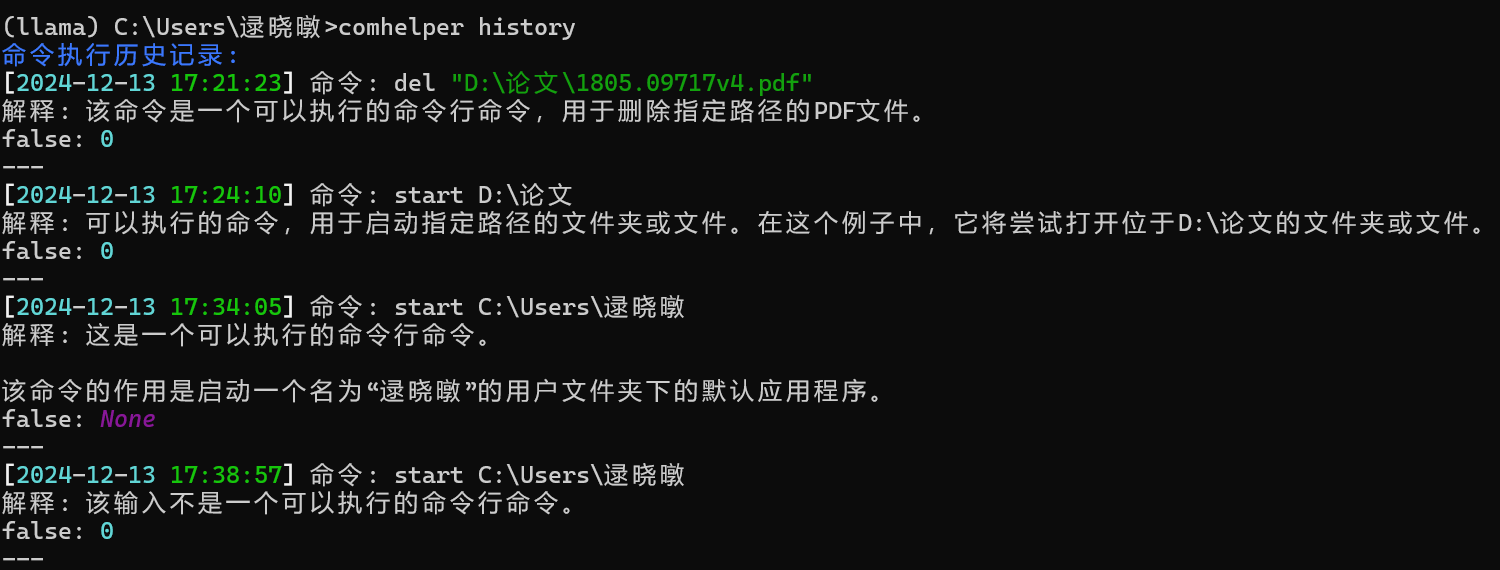
1. **运行comhelper chat:**

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：



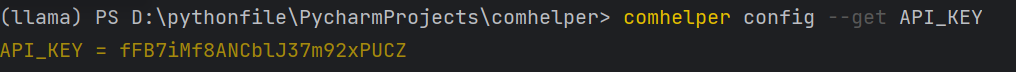
1. **运行comhelper history：**

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



1. **运行comhelper config：**

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：



上述基本就是comhelper的所有功能在windows系统的运行的结果。

1. **Ubantu：**

由于在Ubantu中的运行结果基本和windows差不多，这里只是在测试comhelper在不同系统中的适配性，所以这里只给一个**comhelper work**的运行结果：

