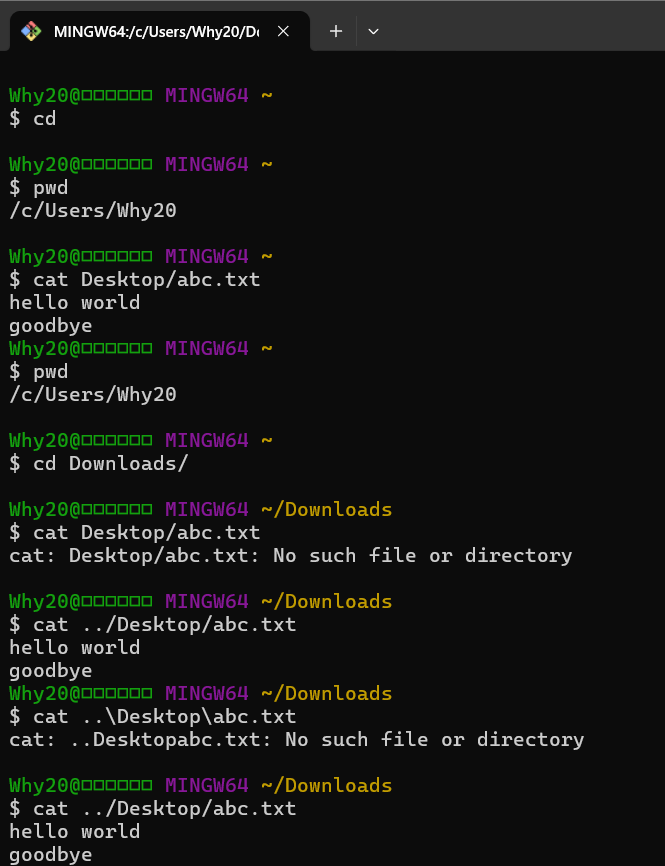
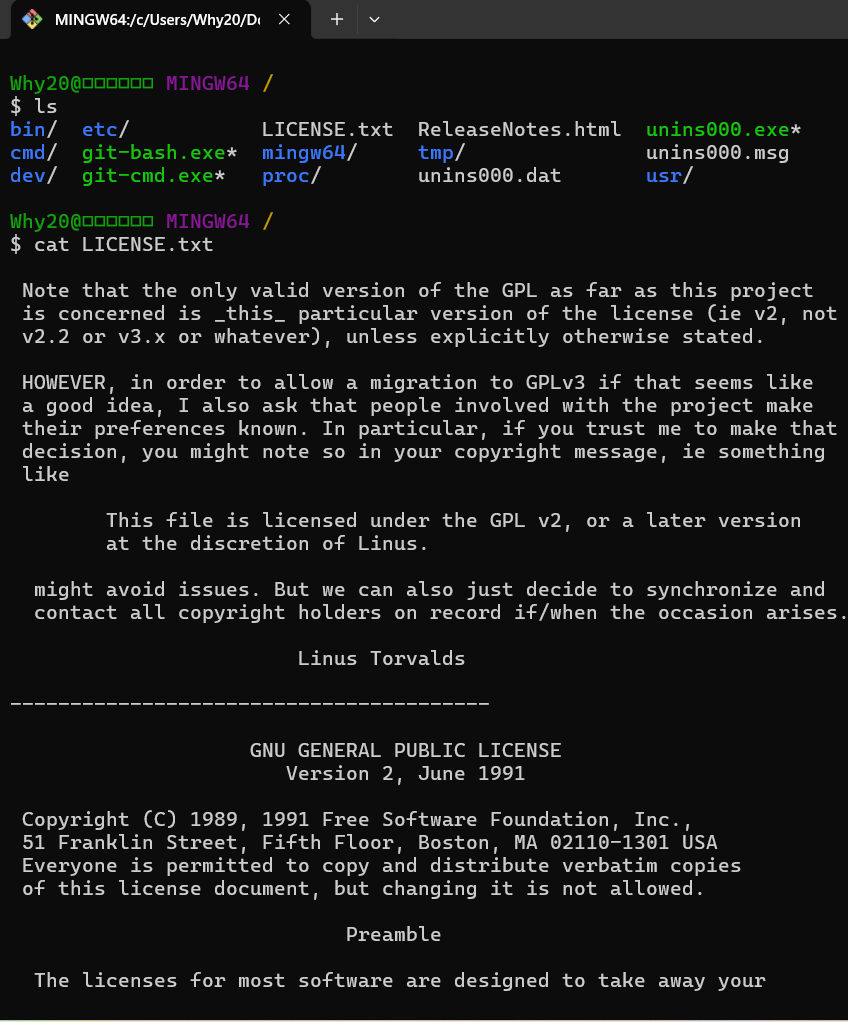
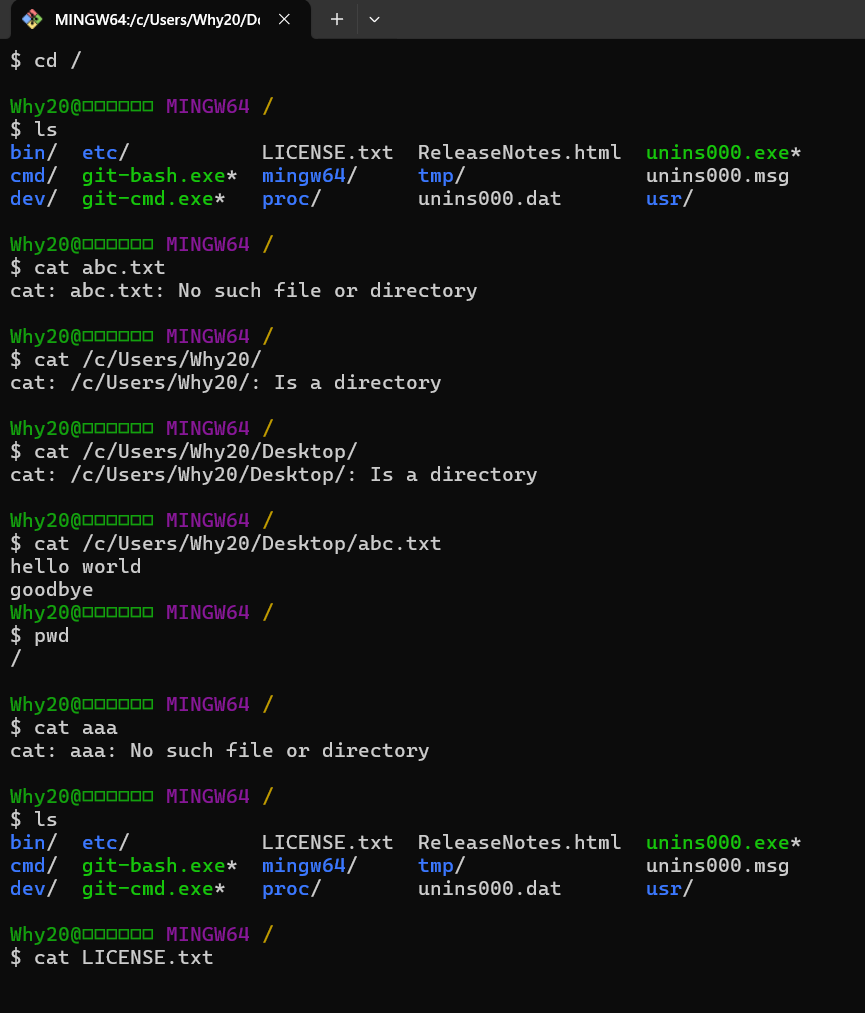
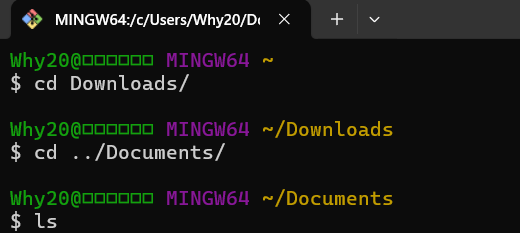
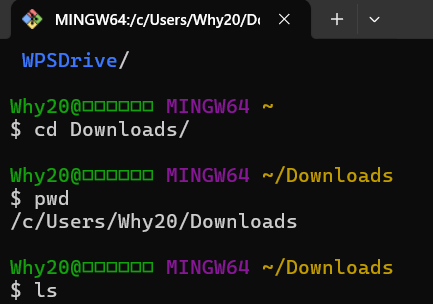
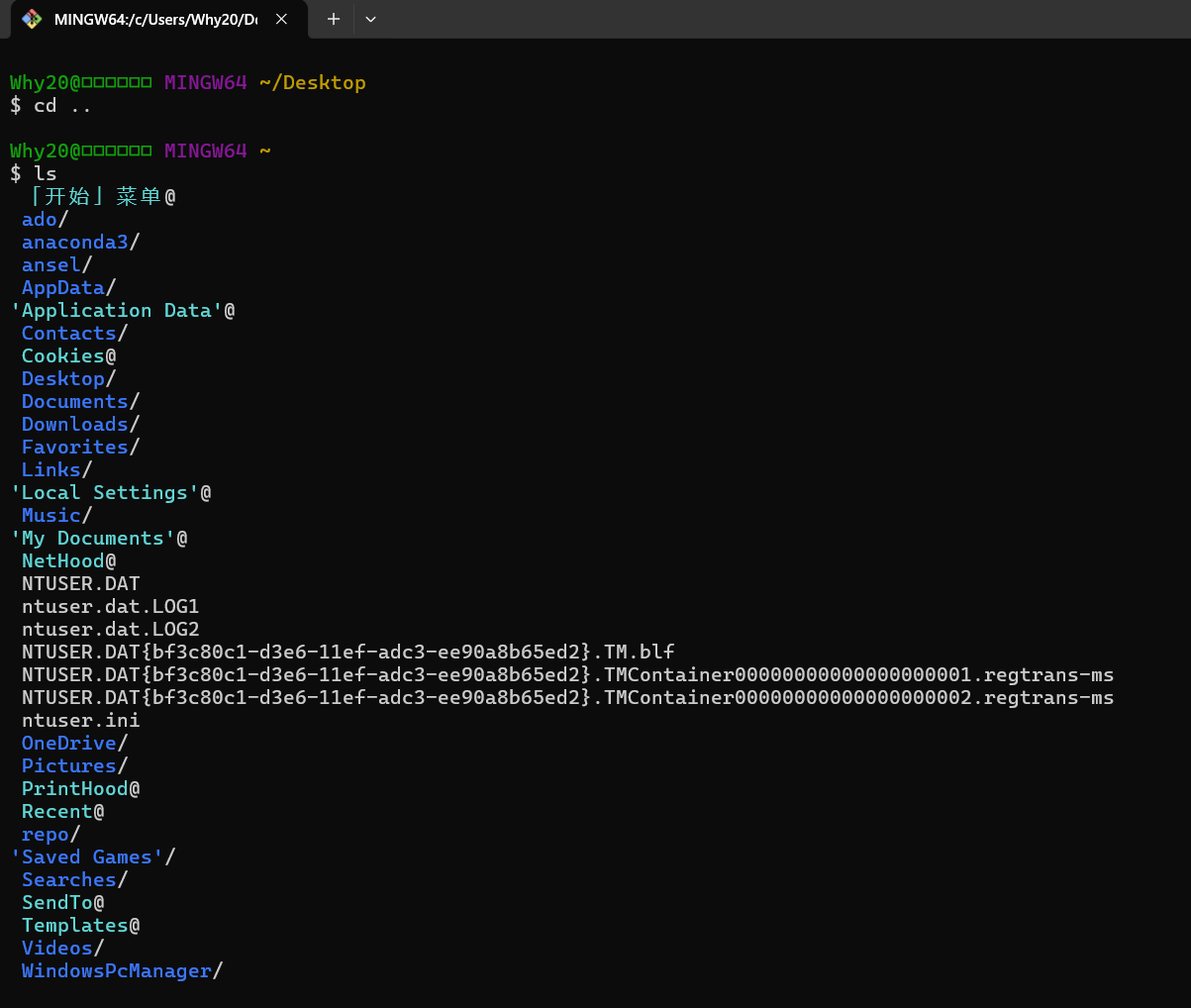
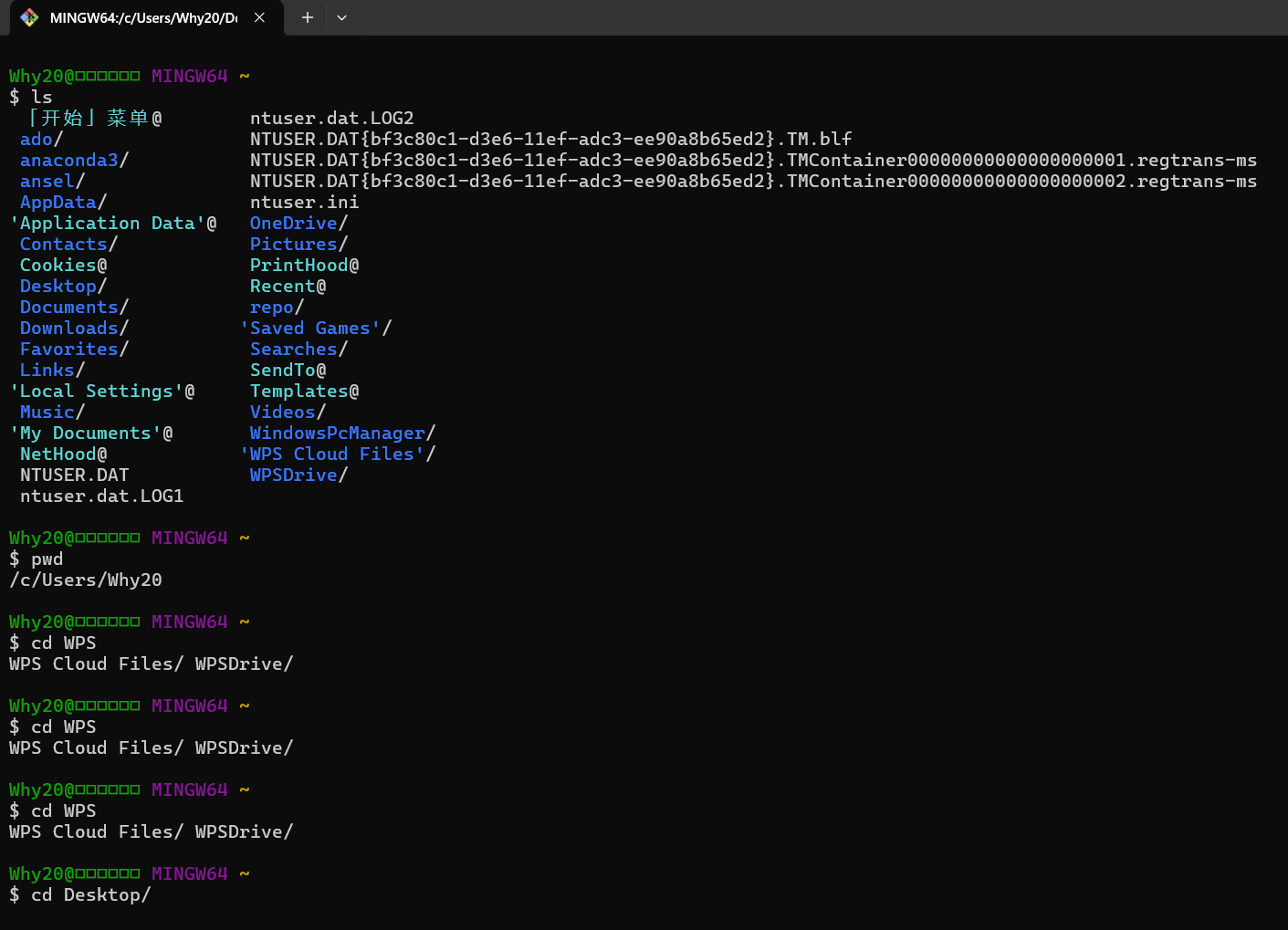
**1.**学习使用ls命令，检查自己计算机最常用的“桌面”、“下载”、“文档”等文件夹的真实的文件系统路径是什么：

①理解“目录树”、“根目录”、“路径”、“绝对路径”、“相对路径”等概念(/分隔符)

②理解“当前工作目录”的概念，掌握pwd命令

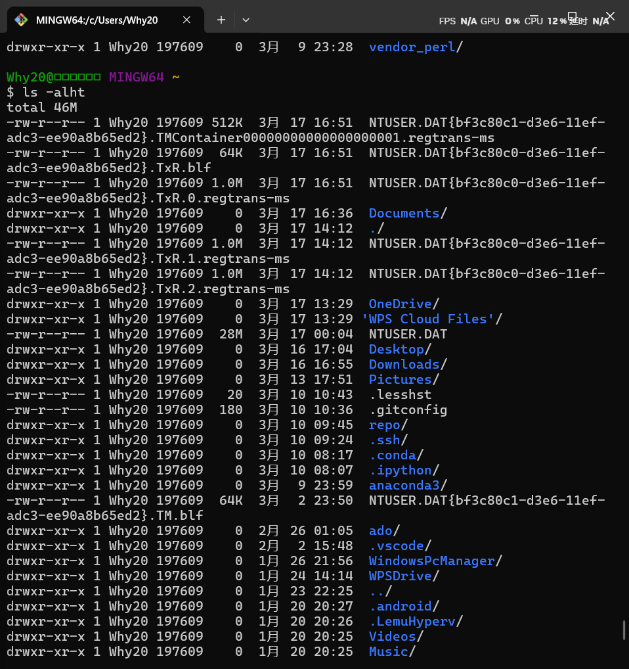
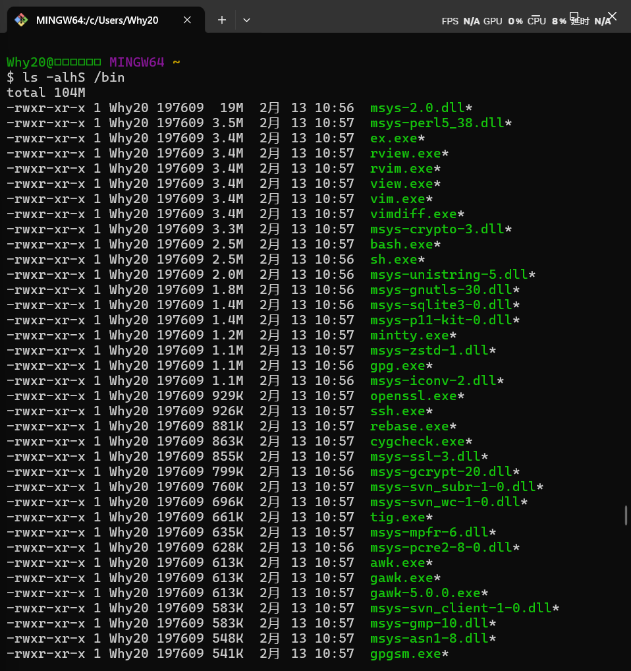
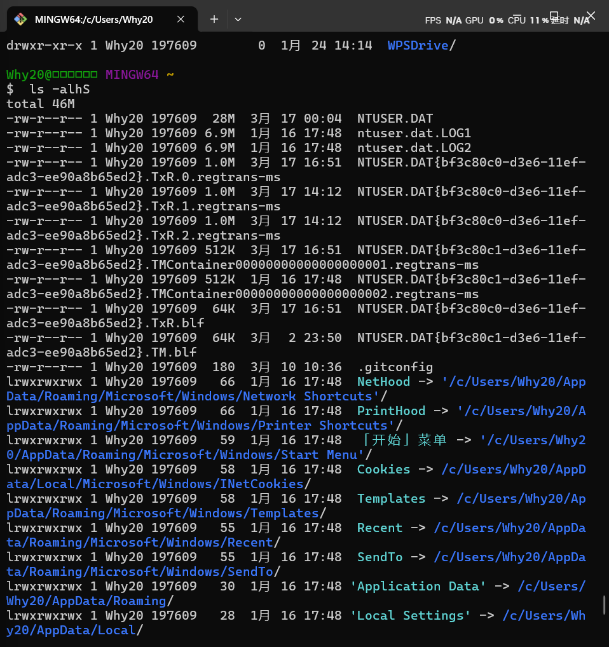
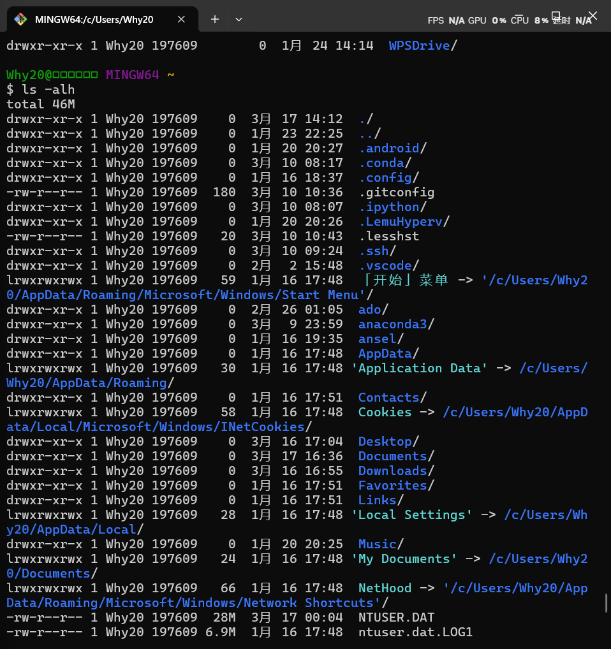
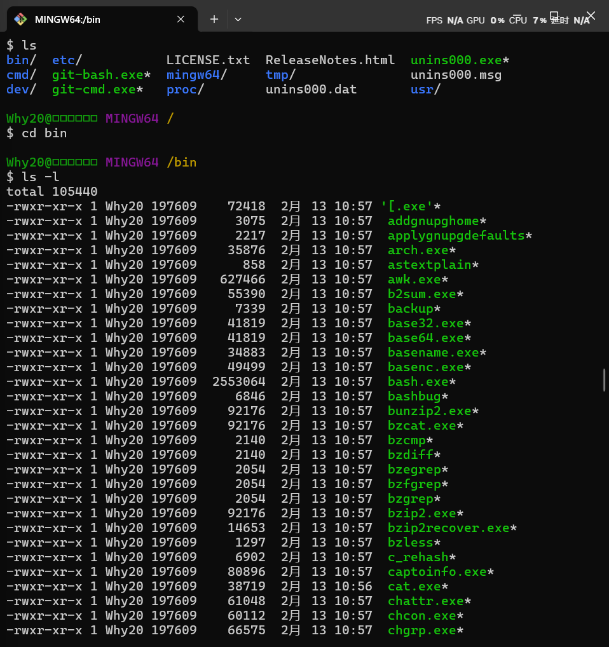
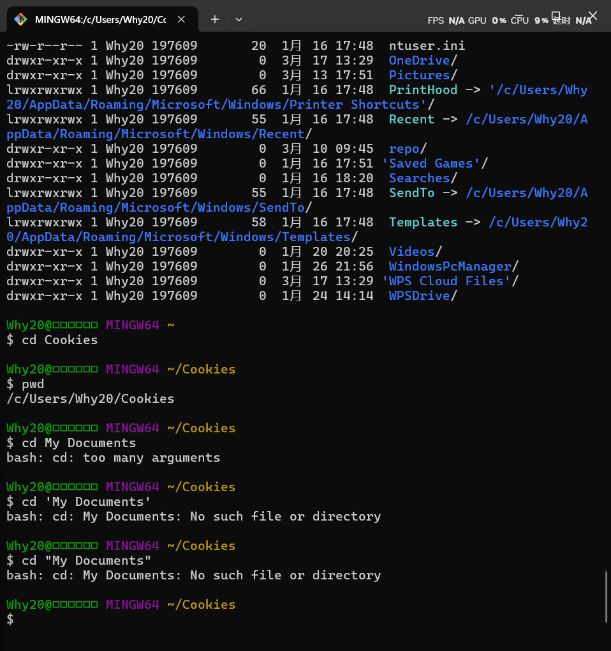
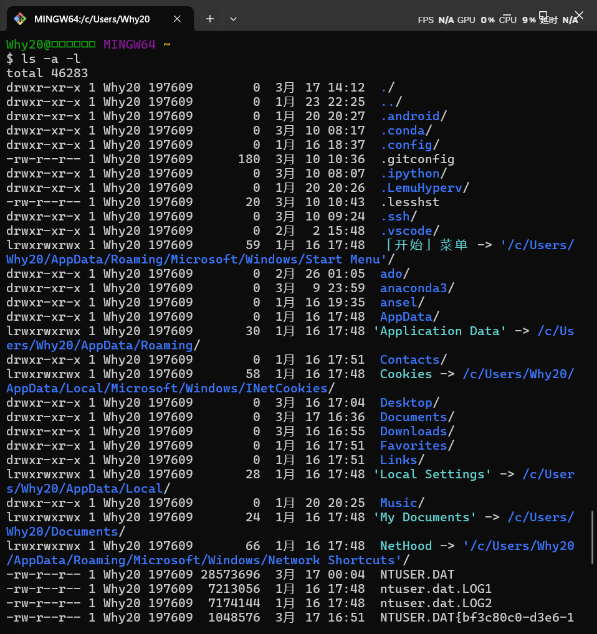
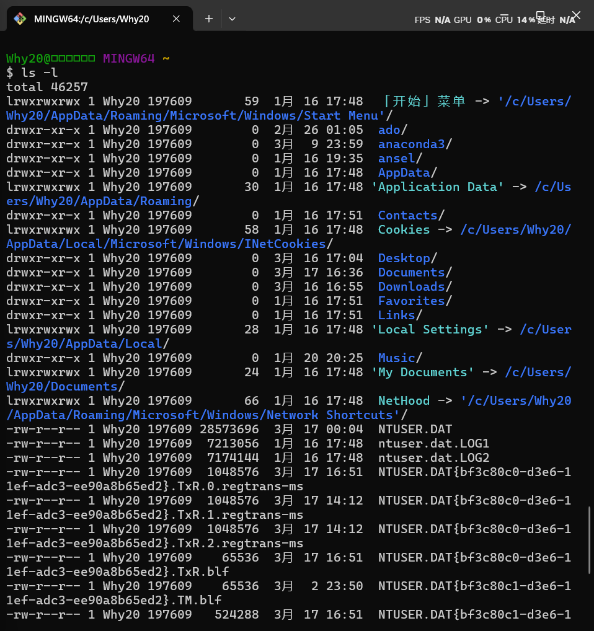
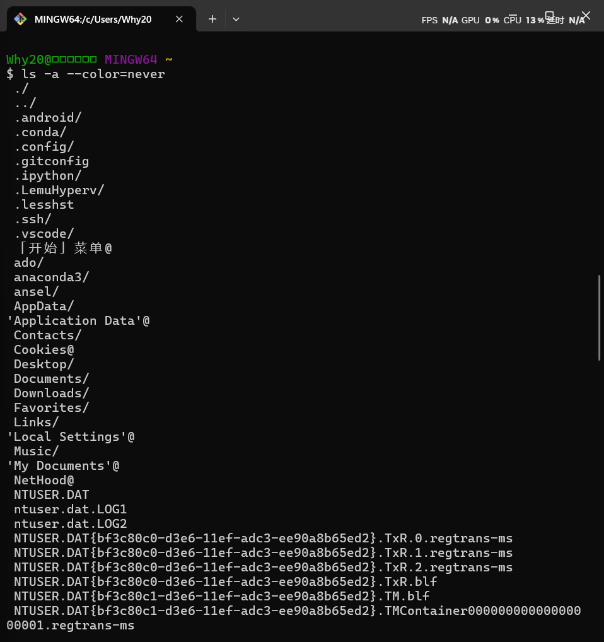
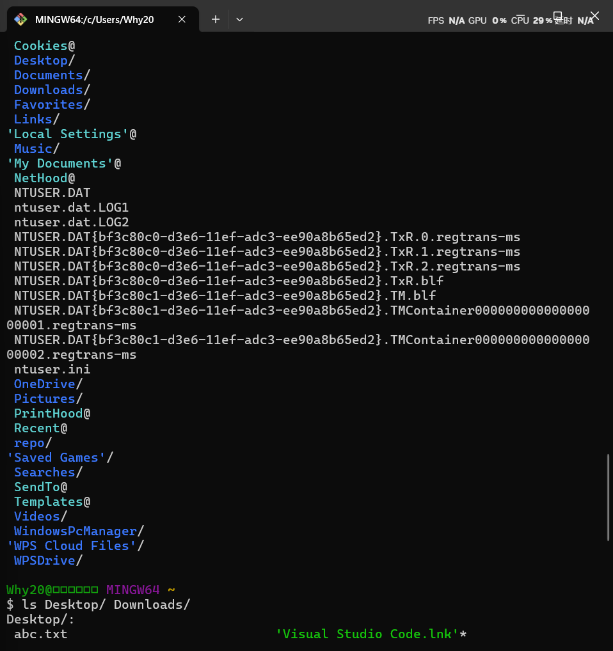
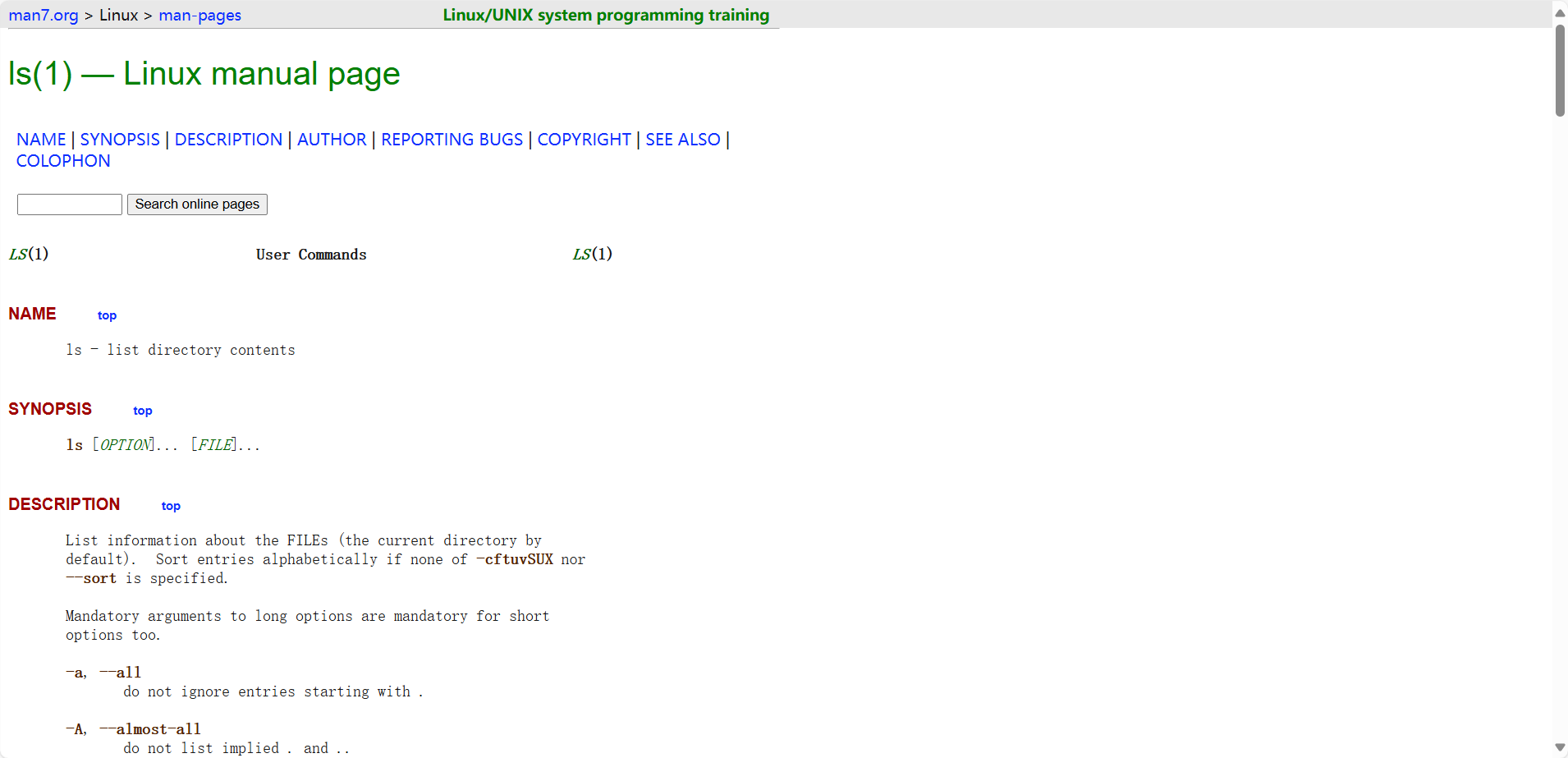
③理解Unix路径的标准写法，以及Windows路径与Unix路径的写法差异



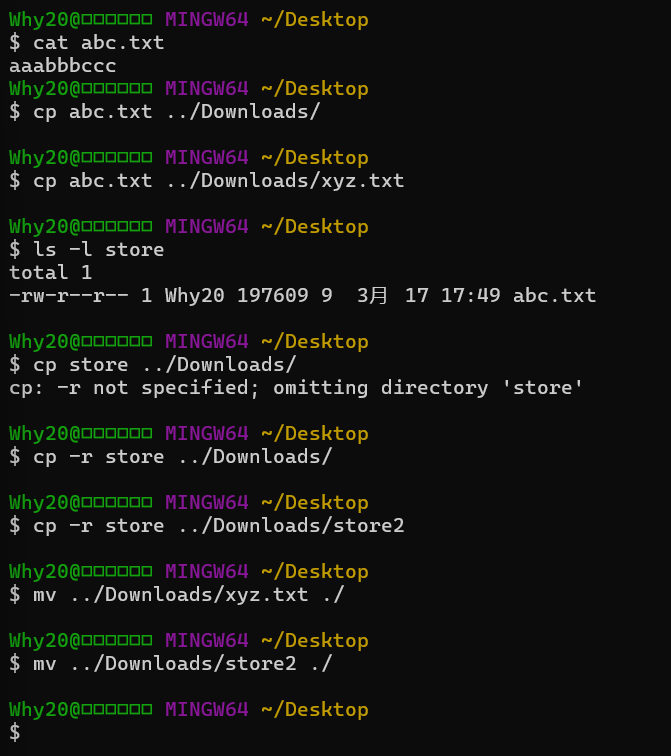
通过ls命令查看"桌面""下载"等文件夹路径，掌握了Unix系统以/分隔的层级结构，如/Users/用户名/Desktop。理解了目录树、绝对路径（从根目录/开始）、相对路径（相对于当前目录）等概念，并用pwd命令实时确认当前工作目录。对比发现Unix路径无盘符且使用/，而Windows以\分隔并需盘符（如C:\）。这些操作让我认识到精准路径管理在文件操作中的重要性，特别是跨平台脚本开发时需注意系统差异，为后续金融编程中的文件处理打下基础。

④理解Shell(Bash、Zsh)的基本语法结构(空格分隔、短选项、长选项、参数)

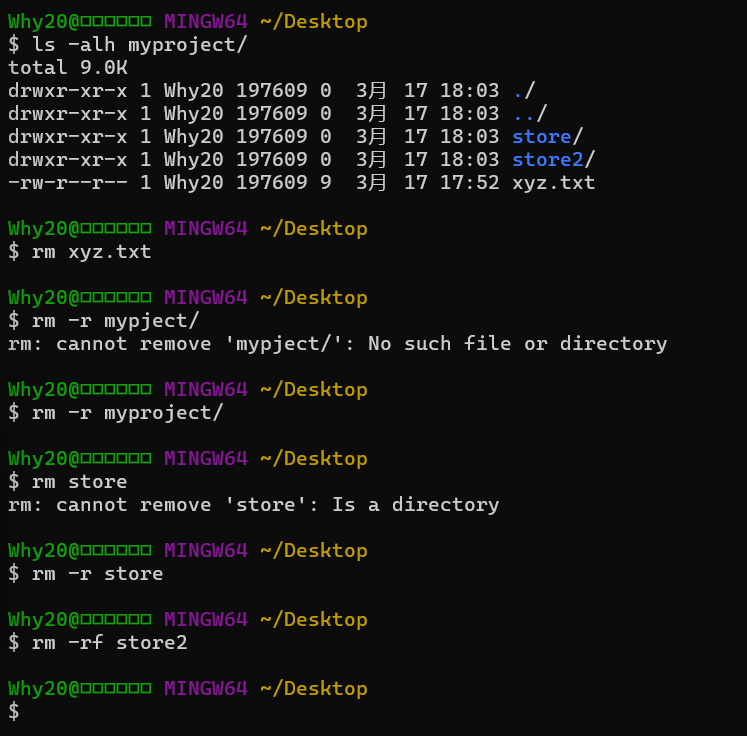
⑤阅读在线版manls手册，也可以使用ls--help命令在终端中显示手册(适用于GitBash和Ubuntu)，尝试ls命令的一些常用选项



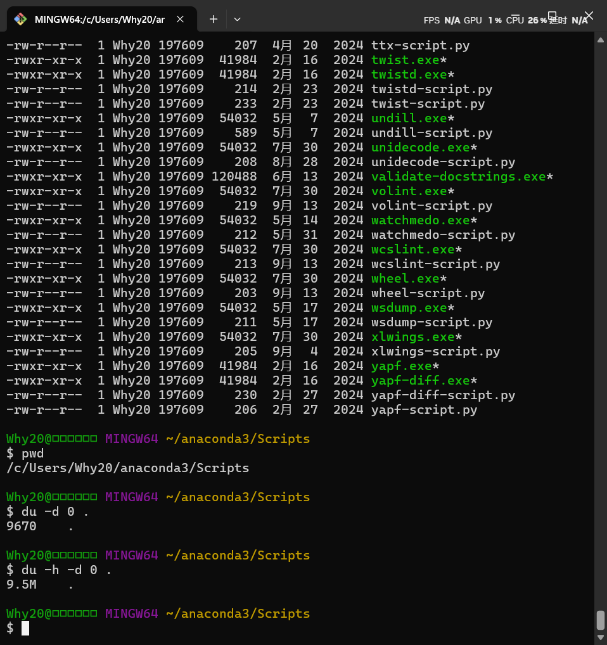
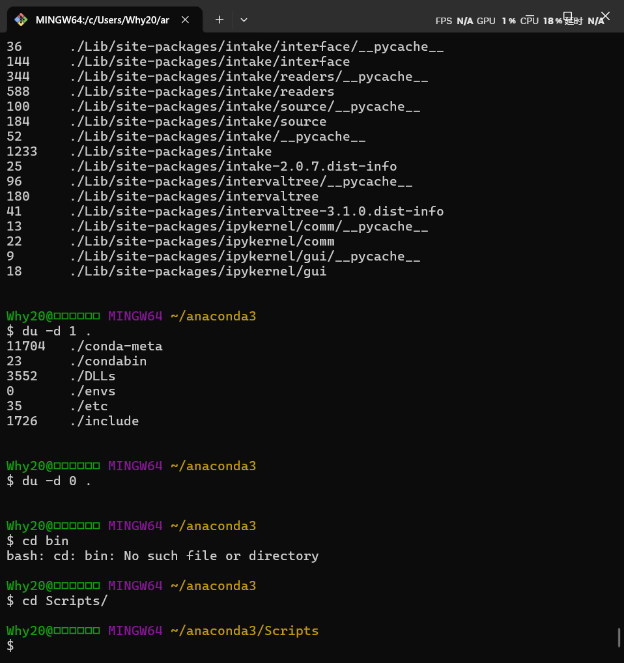
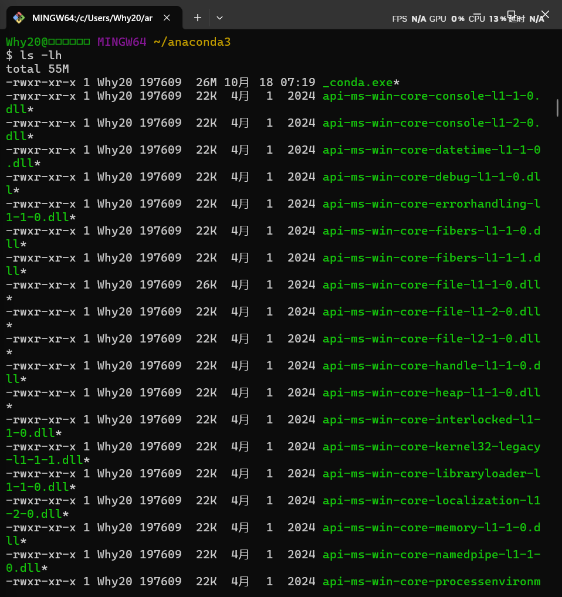
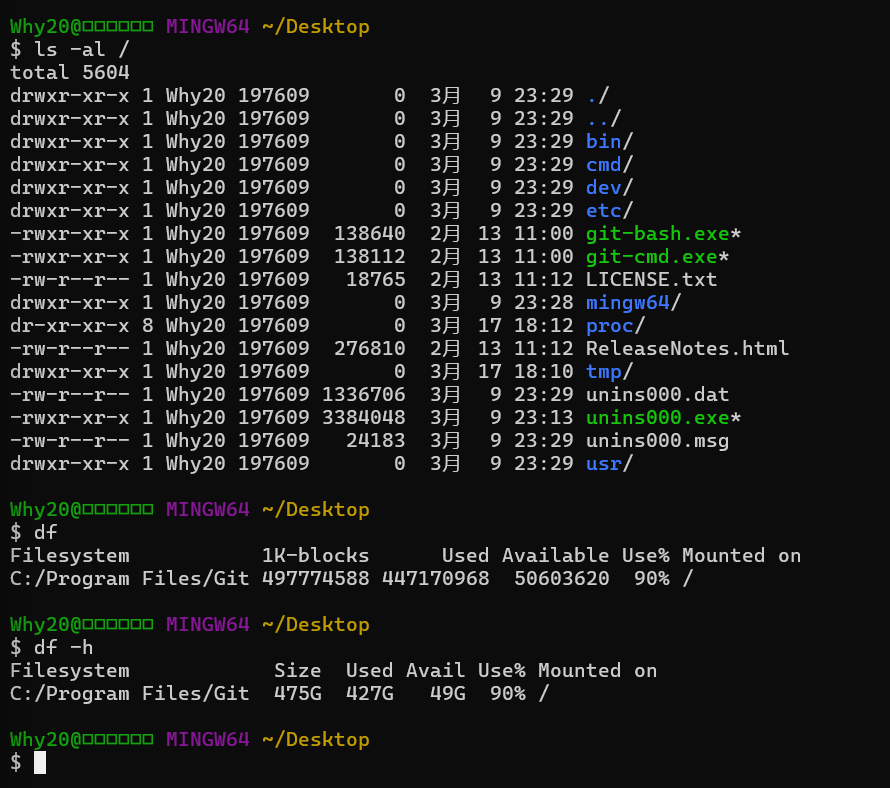
通过实践ls命令的多个选项，我系统掌握了不同参数组合的实际应用场景。ls-a显示所有文件（包括隐藏文件）。ls-al以长格式展示详细信息，包括权限、所有者、大小、修改时间等，如drwxr-xr-x表示目录权限。添加-h选项后（ls-alh），文件大小以用户可读的KB/MB单位显示，更直观。使用-S选项按文件大小排序（ls-alhS）；而-t选项按修改时间排序（ls-alht）能快速定位最新编辑的文档。通过ls-alhS/bin查看系统二进制目录，观察到命令文件的权限特征和大小分布。这些操作实践让我意识到ls命令的灵活性在于选项组合，能根据需求高效过滤和排序文件信息，对金融数据文件的快速检索和管理具有重要实用价值。

**2.**使用cp命令复制文件/文件夹，使用mv命令移动(重命名)文件/文件夹

**3.**使用mkdir命令创建一个名为myproject的新文件夹，然后从图形界面(比如Windows的“文件资源管理器”、macOS的“访达”)复制粘贴某些文件/文件夹进去，用ls命令查看复制进去的文件/文件夹的大小和修改时间，最后用rm命令删除这些文件/文件夹



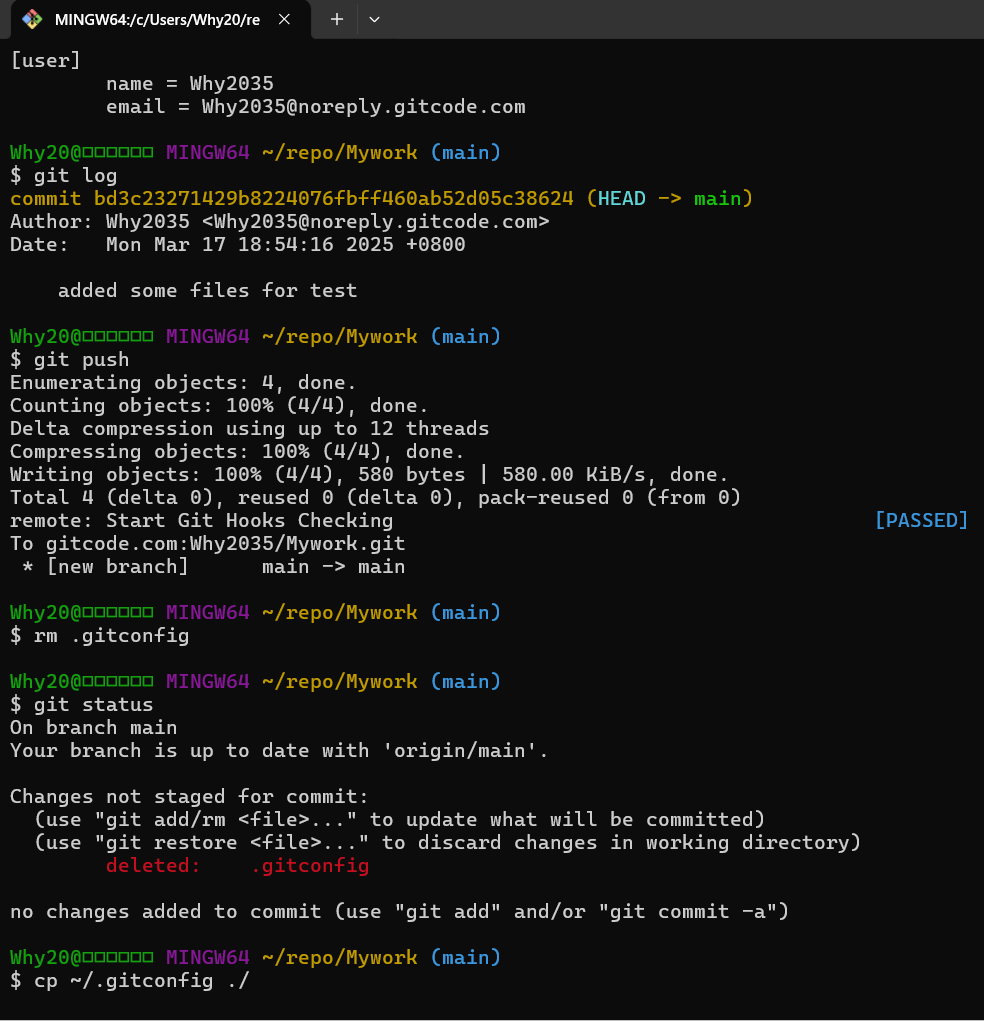
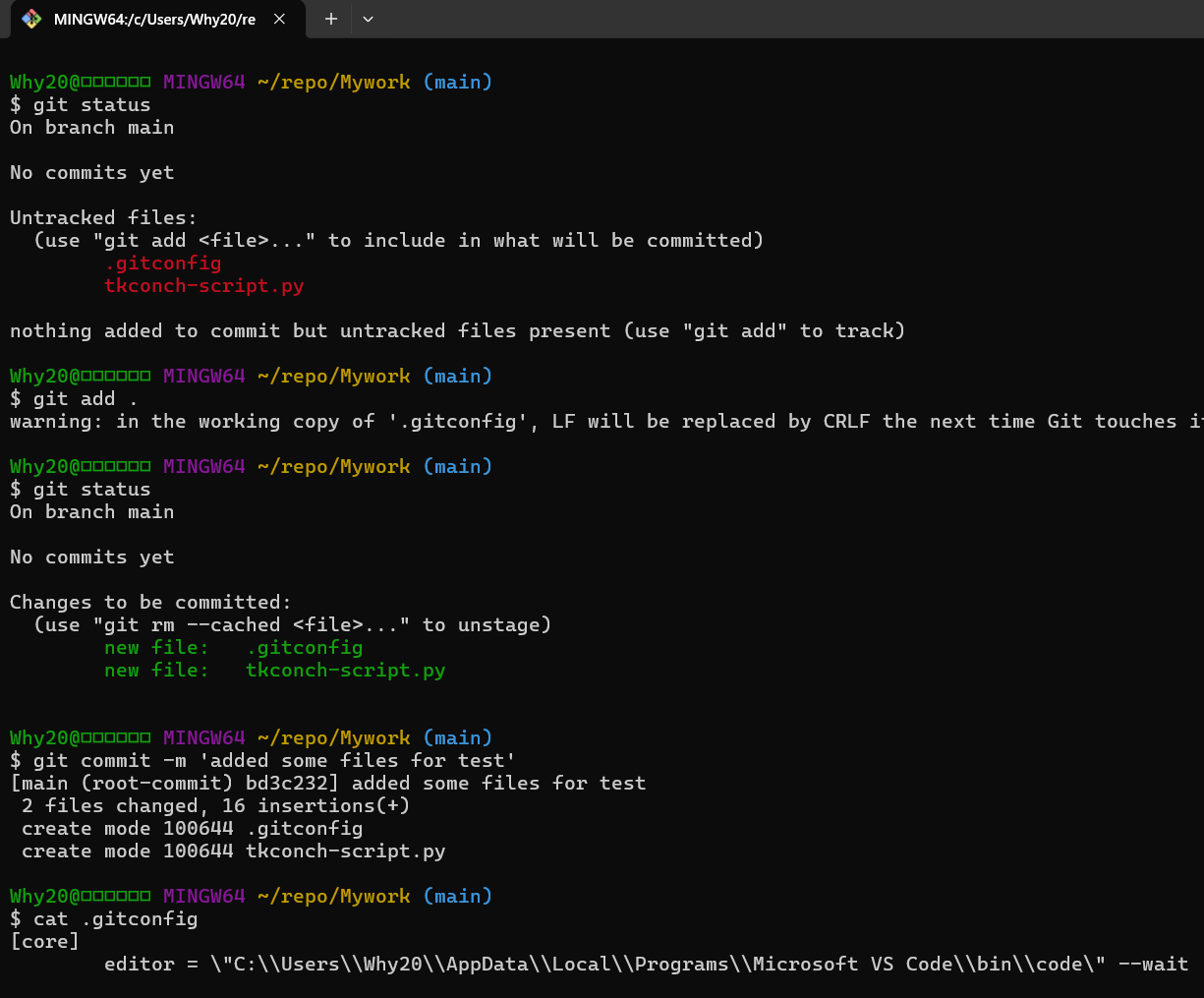
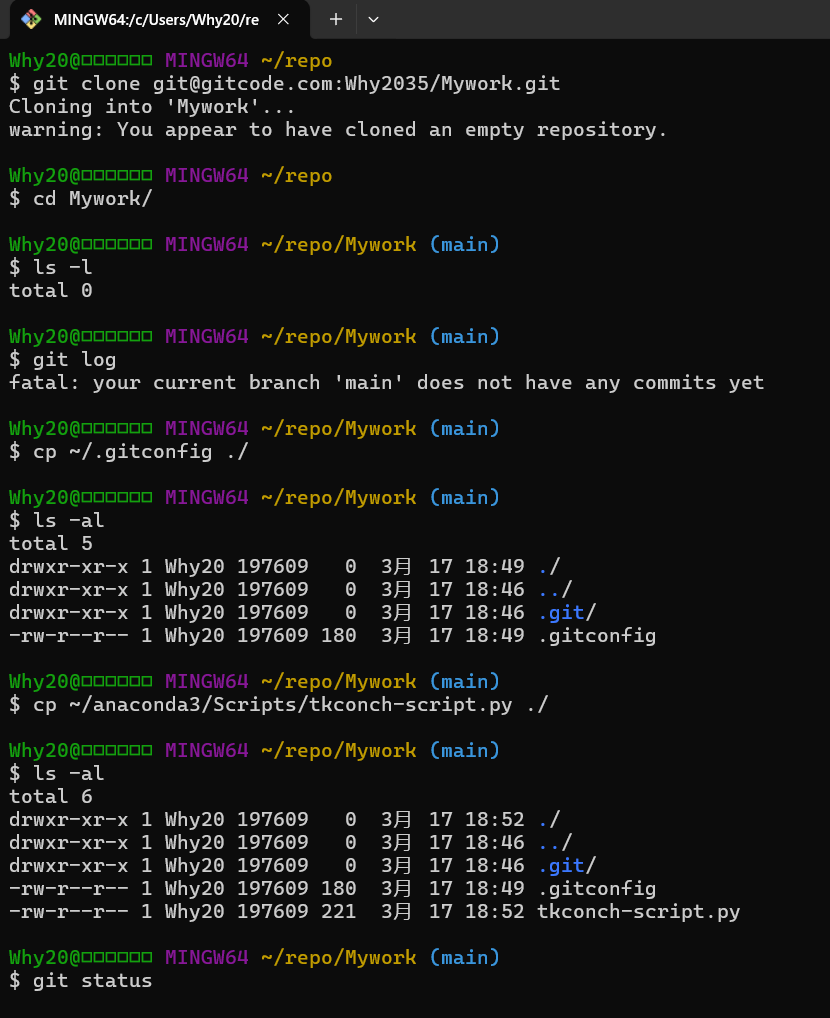
**4.**使用df命令查看磁盘剩余空间，使用du命令查看文件/文件夹占用的磁盘空间



**5.**使用某个AI大模型(比如豆包，或DeepSeek)解释下面的Bash命令，

du-s\*|sort-nr>~/report.txt

**6.**在GitCode平台新建一个你个人的私密的代码仓库(非公开，别人看不到)，clone到本地，将一些你自己的工作文件(文本文件或二进制文件都可以)添加到仓库里，push到平台上托管



这次在GitCode平台创建私密仓库并管理文件的操作让我收获颇丰。新建私密仓库保障了我的代码安全，不用担心信息泄露。将仓库克隆到本地后，操作更加便捷。把工作文件添加到仓库，使用gitadd和gitcommit对文件进行版本管理，让我对文件的修改历史有清晰的记录。最后`gitpush`到平台，实现了代码的远程托管。整个过程让我更熟悉Git操作，提高了工作效率和代码管理能力。