

第二周学习报告

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

- 理解“目录树”、“根目录”、“路径”、“绝对路径”、“相对路径”等概念 (/ 分隔符)
- 理解“当前工作目录”的概念, 掌握 `pwd` 命令
- 理解 Unix 路径的标准写法, 以及 Windows 路径与 Unix 路径的写法差异
- 理解 Shell (Bash、Zsh) 的基本语法结构 (空格分隔、短选项、长选项、参数)
- 阅读在线版 `man ls` 手册, 也可以使用 `ls --help` 命令在终端中显示手册 (适用于 Git Bash 和 Ubuntu), 尝试 `ls` 命令的一些常用选项

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

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

`ls` 当前工作目录下的列表

Tab 自动补全功能。例: 输入 `des` 补全成 `Desktop/` (只有一种可能才会按一次就自动补全, 如果有两种可能, 按两次 Tab 键会把所有的可能都列出来)

提示符显示当前工作目录

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop
```

`cat` 显示文件内容

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop
$ cat abc.txt
hello
```

如果更改文件内容, 重新调用上一个命令语句, 直接按 `↑` 即可

`cd ..` 回到上一级文件夹 `cd .` 当前文件夹

根目录:

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop
$ pwd
/c/Users/李意如/Desktop
```

`cd /` 可以回到根目录去

路径: 在命令行里, 要访问某一个东西的时候, 就需要给出一个路径

绝对路径: 从目录树的根目录开始往下面写

```
李意如@LAPTOP-9J8HOMDD MINGW64 /
$ cat /c/Users/李意如/Desktop/abc.txt
hello
good bye
```

相对路径: 相对于当前工作目录 `Desktop` 下的 `abc.txt` (只要不是从根目录开始写都是相对路径)

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop
$ cat abc.txt
hello
```

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~  
$ cat Desktop/abc.txt  
hello  
good bye
```

/ 根目录或者分隔符

- 理解 “当前工作目录” 的概念，掌握 `pwd` 命令

当前工作目录: `pwd`

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

Unix 路径 “/” 斜杠 slash，根目录为 “/”

```
$ cat /c/Users/李意如/Desktop/abc.txt  
hello  
good bye
```

Windows 路径 “\” 反斜杠 back slash。没有根目录概念，是从盘符开始的，例: “C: ” “D: ”

```
C:\Users\李意如
```

浏览器里是 “/” 的原因: 浏览器在互联网上，互联网是一个开放的标准，遵循 Unix 标准

<https://gitcode.com/turnsolee/courses/blob/main/01-beginner/02.tutorial.md>

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

参数

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/Downloads  
$ cd ..
```

长选项和短选项效果一样，例:

```
-a, --all  
do not ignore entries starting with .
```

`ls` 后的文件夹可以有多个，会分别列举

```
ls [OPTION]... [FILE]...
```

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

`man XX` 都可以阅读

`ls -a` 或者 `ls --all` 显示所有隐藏文件

`ls -l`

`ls -al` 等价于 `ls -a -l`

```

李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ ls -al
total 22264
drwxr-xr-x 1 李意如 197121 0 3月 8 20:50 ./
drwxr-xr-x 1 李意如 197121 0 6月 29 2024 ../
drwxr-xr-x 1 李意如 197121 0 2月 5 2024 .android/
drwxr-xr-x 1 李意如 197121 0 3月 7 16:25 .conda/
-rw-r--r-- 1 李意如 197121 132 3月 6 20:25 .condarc
-rw-r--r-- 1 李意如 197121 90 3月 8 20:50 .gitconfig
drwxr-xr-x 1 李意如 197121 0 2月 19 19:37 .idlerc/
drwxr-xr-x 1 李意如 197121 0 3月 7 16:25 .ipython/
-rw-r--r-- 1 李意如 197121 20 3月 8 19:14 .lessht
drwxr-xr-x 1 李意如 197121 0 3月 7 16:25 .ssh/
drwxr-xr-x 1 李意如 197121 0 3月 7 16:25 .vscode/
lrwxrwxrwx 1 李意如 197121 63 5月 19 2024 [开始] 菜单 -> '/c/Users/李意如/AppData/Roaming/Microsoft/Windows/St

```

第一个字母：“d”代表文件夹，“-”表示普通文件，“l”表示符号链接，可以做链接替换跳转，类似于传送门

后面字母为权限，“r” read “w” write “x” execute 执行

例：“rw-”为可读可写不可运行；“r-x”可读可执行不可写；

一共有三组，第一组在 Unix 的操作系统里面对应的所有者 owner 权限，第二组为用户所在组 group 的权限，第三组为其他人的权限。“李意如”为 owner 所有者 “197121”为组 group。

-h, --human-readable

with **-l** and **-s**, print sizes like 1K 234M 2G etc.

\$ ls -alh “h” 更可读 文件大小更可读

```

-rw-r--r-- 1 李意如 197121 14K 3月 12 00:04 NTUSER.DAT
-rw-r--r-- 1 李意如 197121 3.6K 5月 19 2024 ntuser.dat.LOG1
-rw-r--r-- 1 李意如 197121 3.0K 5月 19 2024 ntuser.dat.LOG2
-rw-r--r-- 1 李意如 197121 64K 3月 7 20:49 NTUSER.DAT{88a05621-15ef-11ef-86da-9766a6d6956d}.TM.blf
-rw-r--r-- 1 李意如 197121 512K 3月 7 20:49 NTUSER.DAT{88a05621-15ef-11ef-86da-9766a6d6956d}.TM.Container000000000000

```

-S 按文件大小进行排序 largest first

-t sort by time, newest first; see **--time**

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

cp 复制文件

```

李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop
$ cp abc.txt ../Downloads/

```

abc.txt	2025/3/16 12:36	文本文档	1 KB
▼ 上周			
20230925102206000752.docx	2025/3/7 19:36	DOCX 文档	166 KB
Anaconda3-2024.10-1-Windows-x86...	2025/3/6 20:07	应用程序	973,336 KB

```

李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop
$ cp abc.txt ../Downloads/xyz.txt

```

同时改了名字

名称	修改日期	类型	大小
▼ 今天			
xyz.txt	2025/3/16 12:38	文本文档	1 KB
abc.txt	2025/3/16 12:36	文本文档	1 KB
▼ 上周			
20230925102206000752.docx	2025/3/7 19:36	DOCX 文档	166 KB

复制文件夹，**recursively** 递归 改名类似

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop
$ cp -r store ../Downloads/
```

-R, --recursive
list subdirectories recursively

mv 相当于剪切，用法和 cp 类似，但是移动文件夹时不需要输入 “-r”

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

删除要谨慎，没有回收站这一说法，要看好路径

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop
$ mkdir myproject

李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop
$ ls -alh myproject/
total 29K
drwxr-xr-x 1 李意如 197121 0 3月 16 13:04 ./
drwxr-xr-x 1 李意如 197121 0 3月 16 13:04 ../
drwxr-xr-x 1 李意如 197121 0 3月 16 12:41 store/
-rw-r--r-- 1 李意如 197121 9 3月 16 12:38 xyz.txt

李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop
$ rm -r myproject/
```

rm -rf 加一个 “f”，强制删除

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

df “disk free”

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop
$ df
Filesystem            1K-blocks      Used Available Use% Mounted on
C:/Program Files/Git 104857596 61255860  43601736  59% /
D:                    393942012 75300996  318641016  20% /d

李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop
$ df -h
Filesystem            Size  Used Avail Use% Mounted on
C:/Program Files/Git 100G   59G   42G   59% /
D:                   376G   72G   304G   20% /d
```

du “disk used”

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop/公司金融
$ du .
6200    ./展示
20036   ./点评
26260   .
```

“ctrl+c”命令的打断

深度为 1

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/Desktop/公司金融
$ du -d 1 .
6200    ./展示
20036   ./点评
26260   .
```

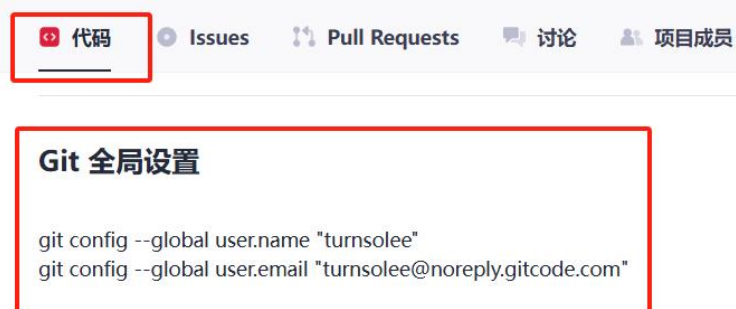
5. 使用某个 AI 大模型 (比如豆包, 或 DeepSeek) 解释下面的 Bash 命令,

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

理解其含义后, 适当修改, 尝试在自己的计算机上应用。

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

全局设置, 只需要进行一次。分开合并配置都可以



分开复制, 进行两次命令即可成功。

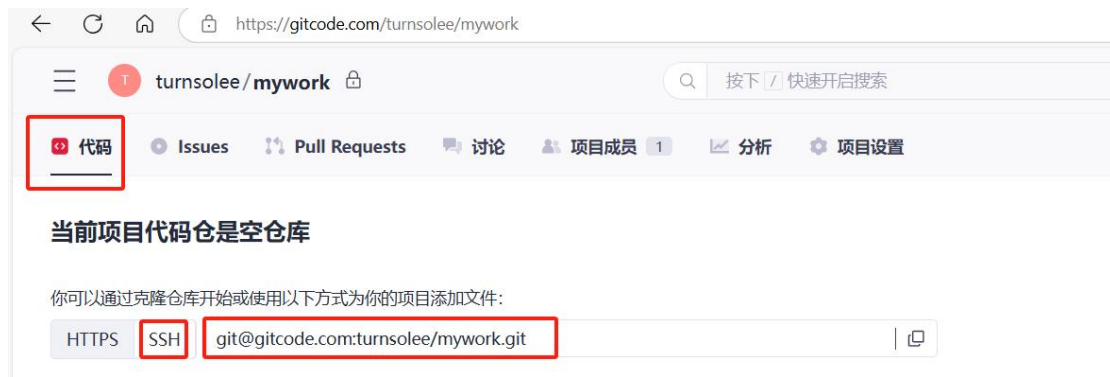
怎么查看 `cat ~/.gitconfig`

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo/mywork (main)
$ cat ~/.gitconfig
[user]
    name = turnsolee
    email = turnsolee@noreply.gitcode.com
[core]
    quotepath = false
```

`git config --list --global`

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo/mywork (main)
$ git config --list --global
user.name=turnsolee
user.email=turnsolee@noreply.gitcode.com
core.quotepath=false
```

开始克隆仓库



```
李意如@LAPTOP-9J8HOMDD MINGW64 ~  
$ cd repo  
  
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo  
$ pwd  
/c/Users/李意如/repo  
  
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo  
$ git clone git@gitcode.com:turnsolee/mywork.git  
Cloning into 'mywork'...  
warning: You appear to have cloned an empty repository.
```

`git log` 是 Git 版本控制系统中的一个非常常用的命令，用于查看项目的提交历史记录。

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo/mywork (main)  
$ git log  
fatal: your current branch 'main' does not have any commits yet
```

复制 `environment.txt` 到当前文件夹。~表示用户主目录，比如用户“李意如”的主目录就是“/Users/ 李意如”

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo/mywork (main)  
$ cp ~/.conda/environments.txt ./
```

这些文件已经复制到了私人主仓库

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo/mywork (main)  
$ ls -al  
total 5  
drwxr-xr-x 1 李意如 197121 0 3月 16 14:29 ./  
drwxr-xr-x 1 李意如 197121 0 3月 16 14:05 ../  
drwxr-xr-x 1 李意如 197121 0 3月 16 14:05 .git/  
-rw-r--r-- 1 李意如 197121 23 3月 16 14:29 environments.txt
```

`git status` 是 Git 版本控制系统里一个极为常用的命令，其主要功能是查看当前工作目录和暂存区的状态，

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo/mywork (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        environments.txt

nothing added to commit but untracked files present (use "git add" to track)
```

该文件尚未纳入版本控制，游离在代码仓库以外

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo/mywork (main)
$ git add .
```

“.” 当前文件夹

再次查看状态

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo/mywork (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   environments.txt
```

这是执行 `git status` 命令后的输出，显示当前处于 `main` 分支，且还没有任何提交记录。“Changes to be committed” 部分表明文件 `environments.txt` 已被暂存，即使用 `git add` 命令将其添加到了待提交状态。

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo/mywork (main)
$ git commit -m "为了练手随便添加的"
[main (root-commit) 071a0b2] 为了练手随便添加的
1 file changed, 2 insertions(+)
create mode 100644 environments.txt
```

`git log` 查看提交历史

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo/mywork (main)
$ git log
commit 071a0b227f949ebf79745bb6821512c1645d993f (HEAD -> main)
Author: turnsolee <turnsolee@noreply.gitcode.com>
Date:   Sun Mar 16 14:48:56 2025 +0800

    为了练手随便添加的
```

`git push` 命令后的输出，表明将本地 `main` 分支的内容推送到远程仓库

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo/mywork (main)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 271 bytes | 135.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Start Git Hooks Checking
To gitcode.com:turnsolee/mywork.git
* [new branch]      main -> main
```

[PASSED]

完成



提交作业的时候一定要注意文件夹位置

```
李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo/mywork (main)
$ cd

李意如@LAPTOP-9J8HOMDD MINGW64 ~
$ cd repo

李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo
$ pwd
/c/Users/李意如/repo

李意如@LAPTOP-9J8HOMDD MINGW64 ~/repo
$ git clone git@gitcode.com:turnsolee/week02.git
```