

# 金融编程与计算

## Week2——命令行与文件系统（初级）

一、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

### （一）cd、pwd、ls、cat

```
MINGW64:/c/Users/Administrator

Administrator@PC-20220115WOJX MINGW64 ~
$ ls
AppData/
'Application Data'@
Contacts/
Cookies@
Desktop/
Documents/
Downloads/
Favorites/
Links/
'Local Settings'@
Music/
'My Documents'@
NTUSER.DAT
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM.blf
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer00000000000000000000
1.regtrans-ms
NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer00000000000000000000
2.regtrans-ms
NetHood@
OneDrive/
Pictures/
PrintHood@
Recent@
'Saved Games' /
Searches/
SendTo@
Templates@
Videos/
'WPS Cloud Files' /
WPSDrive/
ado/
ntuser.dat.LOG1
ntuser.dat.LOG2
ntuser.ini
repo/
wc/
「开始」菜单@
下载.lnk*
图片.lnk*
文档.lnk*

Administrator@PC-20220115WOJX MINGW64 ~
$ pwd
/c/Users/Administrator

Administrator@PC-20220115WOJX MINGW64 ~
$
```

注：cd=change directory 改变目录

Pwd=Print Working Directory 显示当前所在的目录路径

ls=List 的缩写，列出当前目录或指定目录的文件和文件夹。

ls -l : Linux/Unix 系统中用于以「详细列表格式」显示目录内容的命令

cd .. 回到上一级  
cd . 当前文件夹  
“/” 是根目录  
“cd /” 到文件系统的根目录

```
Administrator@PC-20220115WOJX MINGW64 ~/Desktop
$ cd /

Administrator@PC-20220115WOJX MINGW64 /
$ ls
LICENSE.txt      cmd/  git-bash.exe*  proc/  unins001.exe*
ReleaseNotes.html dev/  git-cmd.exe*   tmp/   unins001.msg
bin/             etc/  mingw64/       unins001.dat  usr/

Administrator@PC-20220115WOJX MINGW64 /
```

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)

绝对路径：从根目录开始写的

```
$ cat /c/Users/Administrator/Desktop/abc.txt
```

相对路径：前面不用斜杠开头（不是从根目录开始写的）

```
$ cat LICENSE.txt
```

```
$ cat Desktop/abc.txt
```

写相对路径时要注意所处的当前工作目录在哪里，

因为相对路径它是取决于当前工作目录的任何的一个程序进程

如下：cat ../Desktop/abc.txt

```
Administrator@PC-20220115WOJX MINGW64 ~/Downloads
$ cd Downloads/abc.txt
bash: cd: Downloads/abc.txt: No such file or directory

Administrator@PC-20220115WOJX MINGW64 ~/Downloads
$ cat:Desktop/abc.txt
bash: cat:Desktop/abc.txt: No such file or directory

Administrator@PC-20220115WOJX MINGW64 ~/Downloads
$ cat ../Desktop/abc.txt
hello zhurui
```

## 2. 理解 Unix 路径的标准写法

Unix 分隔符是斜杠/

windows 分隔符是反斜杠 : backslash

windows 没有根目录的概念，它是从盘符开始的，即“c:”

注：浏览器的网址是互联网标准，遵循 unix

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

输入 `ls -al`

d 开头：代表文件夹

- 开头：普通文件

l 开头：符号链接 (快捷方式)

`ls -alh`

```
Administrator@PC-20220115WOJX MINGW64 ~
$ ls -al
total 28104
drwxr-xr-x 1 Administrator 197121      0 Mar 13 21:46 ./
drwxr-xr-x 1 Administrator 197121      0 Jan 15  2022 ../
drwxr-xr-x 1 Administrator 197121      0 Mar 10 00:21 .anaconda/
-rw-r--r-- 1 Administrator 197121    952 Mar 13 21:30 .bash_history
drwxr-xr-x 1 Administrator 197121      0 Mar 14 16:14 .conda/
-rw-r--r-- 1 Administrator 197121    146 Mar 10 00:22 .condarc
drwxr-xr-x 1 Administrator 197121      0 Mar 10 00:21 .continuum/
-rw-r--r-- 1 Administrator 197121    193 Mar 10 02:52 .gitconfig
drwxr-xr-x 1 Administrator 197121      0 Mar 10 00:34 .ipython/
-rw-r--r-- 1 Administrator 197121     20 Mar 10 03:08 .lessht
drwxr-xr-x 1 Administrator 197121      0 Oct 21  2022 .local/
drwxr-xr-x 1 Administrator 197121      0 Mar 10 01:40 .ssh/
drwxr-xr-x 1 Administrator 197121      0 Mar 10 11:23 .suwell/
drwxr-xr-x 1 Administrator 197121      0 Mar  9 17:39 .vscode/
drwxr-xr-x 1 Administrator 197121      0 Jan  9 11:34 .xp2p/
drwxr-xr-x 1 Administrator 197121      0 Oct 28  2022 AppData/
lrwxrwxrwx 1 Administrator 197121     38 Jan 15  2022 'Application Data' -> /c
/Users/Administrator/AppData/Roaming/
drwxr-xr-x 1 Administrator 197121      0 Jan 15  2022 Contacts/
lrwxrwxrwx 1 Administrator 197121     66 Jan 15  2022 Cookies -> /c/Users/Adm
inistrator/AppData/Local/Microsoft/Windows/INetCookies/
drwxr-xr-x 1 Administrator 197121      0 Mar 14 15:49 Desktop/
drwxr-xr-x 1 Administrator 197121      0 Mar 14 09:29 Documents/
drwxr-xr-x 1 Administrator 197121      0 Mar 14 14:47 Downloads/
drwxr-xr-x 1 Administrator 197121      0 Jan 17  2022 Favorites/
```

(二) 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

## 1.复制 cp

```
Administrator@PC-20220115W0JX MINGW64 ~/Downloads
$ cd

Administrator@PC-20220115W0JX MINGW64 ~
$ pwd
/c/Users/Administrator

Administrator@PC-20220115W0JX MINGW64 ~
$ cd Desktop/

Administrator@PC-20220115W0JX MINGW64 ~/Desktop
$ ld -l
bash: ld: command not found

Administrator@PC-20220115W0JX MINGW64 ~/Desktop
$ ls -l
total 53355
-rw-r--r-- 1 Administrator 197121    40864 Dec 25 10:55 '02 赛宝认证公
```

cp 命令复制——cp abc.txt ../Downloads （在上一级 ..复制到新路径）

```
Administrator@PC-20220115W0JX MINGW64 ~/Desktop
$ cp abc.txt ../Downloads/

Administrator@PC-20220115W0JX MINGW64 ~/Desktop
$
```

复制并改名——cp abc.txt ../Downloads/xyz.txt

```
Administrator@PC-20220115W0JX MINGW64 ~/Desktop
$ cp abc.txt ../Downloads/

Administrator@PC-20220115W0JX MINGW64 ~/Desktop
$ cp abc.txt ../Downloads/xyz.txt
```

cp: 复制文件

cp -r: 复制文件夹（eg :\$ cp -r store ../Downloads/）改名同上

## 2.移动 mv

\$ mv ../Downloads/xyz.txt ./

```
Administrator@PC-20220115WOJX MINGW64 ~/Desktop
$ mv ../Downloads/xyz.txt ./

Administrator@PC-20220115WOJX MINGW64 ~/Desktop
$ mv ../Downloads/store2 ./
```

注：mv 不用-r，因为 move 没有递归的算法

（三）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

1. mkdir 命令创建一个名为 myproject 的新文件夹

（注：mkdir=make directory）

```
Administrator@PC-20220115WOJX MINGW64 ~/Desktop
$ mkdir myproject
```

2. 把 store2 复制进 myproject

3. 用 ls 命令查看复制进去的文件/文件夹的大小和修改时间

ls -alh myproject

```
Administrator@PC-20220115WOJX MINGW64 ~/Desktop
$ ls -alh myproject
total 41K
drwxr-xr-x 1 Administrator 197121 0 Mar 14 15:20 ./
drwxr-xr-x 1 Administrator 197121 0 Mar 14 15:11 ../
drwxr-xr-x 1 Administrator 197121 0 Mar 14 15:19 store/
drwxr-xr-x 1 Administrator 197121 0 Mar 14 15:19 store2/
-rw-r--r-- 1 Administrator 197121 5 Mar 14 14:16 xyz.txt
```



- **ls**: 基础命令, 列出目录内容。
- **-a**: 显示 **所有文件** (包括以 **.** 开头的隐藏文件) 。
- **-l**: 以 **长格式** 显示详细信息 (权限、所有者、修改时间等) 。
- **-h**: 以 **人类可读的单位** 显示文件大小 (如 KB、MB) 。

4. 用 **rm** 命令删除这些文件/文件夹

复制 **cp** 和删除 **rm** 需要递归, 移动不需要递归

**rm -r** : 删除

**rm -rf**: 强制删除

(四) 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
Filesystem                1K-blocks      Used Available Use% Mounted on
C:                        104857596  84356136  20501460   81% C:
D:/Documents/首经贸/作业/金融编程与计算/Git 393942012 103298752 290643260   27%
Administrator@PC-20220115WOJX MINGW64 ~/Desktop
$ ls -al
```

**df -h**

```
Administrator@PC-20220115WOJX MINGW64 /d/anaconda/Scripts
$ du -d0 .
10370 .
```

**du -d0 .**

```
Administrator@PC-20220115WOJX MINGW64 /d/anaconda/Scripts
$ du -h -d0 .
11M .
```

**du -h -d0 .**

```
Administrator@PC-20220115W0JX MINGW64 /d/anaconda/Scripts
$ du -h -d0 .
11M      .

Administrator@PC-20220115W0JX MINGW64 /d/anaconda/Scripts
$ du -hs
11M      .

Administrator@PC-20220115W0JX MINGW64 /d/anaconda/Scripts
$ .....
```

（五）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
```

#### 1. `du -s *`

- `du` 是 `disk usage` 的缩写，用于估算文件或目录所占用磁盘空间的大小。
- `-s` 是 `du` 命令的一个选项，即 `--summarize` 的缩写，它的作用是只显示每个参数的总计信息，也就是只输出总大小，而不会像默认情况那样显示每个子目录和文件的详细大小信息。
- `*` 是通配符，表示当前目录下的所有文件和子目录。所以 `du -s *` 会计算当前目录下每个文件和子目录占用磁盘空间的总大小，并将结果输出。

#### 2. `|`（管道符号）

管道符号 `|` 的作用是将前一个命令（`du -s *`）的输出作为后一个命令（`sort -nr`）的输入。

#### 3. `sort -nr`

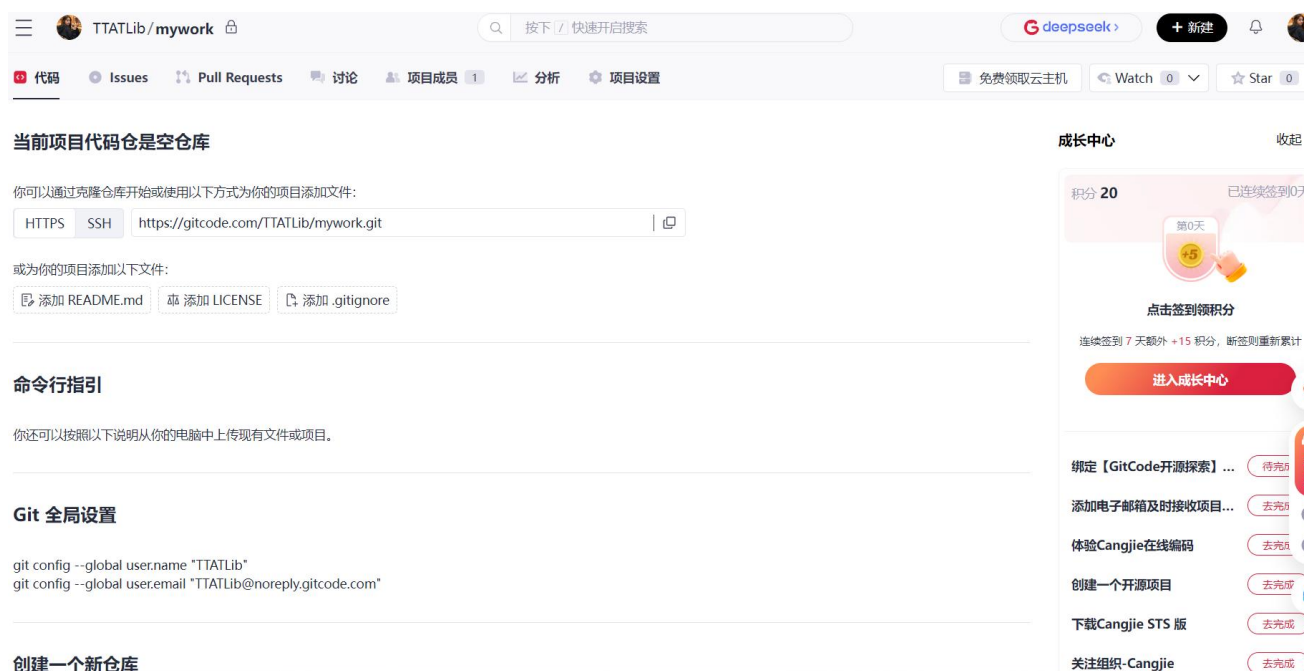
- `sort` 命令用于对文本行进行排序。
- `-n` 选项表示按照数值大小进行排序，而不是按照字母顺序排序。
- `-r` 选项表示以逆序（从大到小）的方式进行排序。所以 `sort -nr` 会对 `du -s *` 输出的磁盘使用量进行数值大小的逆序排序。

#### 4. `> ~/report.txt`

- `>` 是重定向符号，它的作用是将前面命令（`sort -nr`）的输出结果覆盖写入到指定的文件中。
- `~/` 表示用户的主目录。所以 `> ~/report.txt` 会将排序后的磁盘使用情况信息覆盖写入到用户主目录下的 `report.txt` 文件中。如果该文件不存在，则会创建一个新的文件；如果文件已经存在，原文件内容会被新的内容覆盖。

(六) 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

## 1. 创建: 新建——新建项目



## 2. 克隆到本地

(1) 复制 ssh

(2) cd repo

(3) git clone [git@gitcode.com:TTATLib/mywork.git](https://gitcode.com/TTATLib/mywork.git)

```
Administrator@PC-20220115W0JX MINGW64 ~/repo
$ cd mywork/

Administrator@PC-20220115W0JX MINGW64 ~/repo/mywork (main)
$ ls -l
total 0

Administrator@PC-20220115W0JX MINGW64 ~/repo/mywork (main)
$ git long
git: 'long' is not a git command. See 'git --help'.

The most similar commands are
  clone
  log

Administrator@PC-20220115W0JX MINGW64 ~/repo/mywork (main)
$ git log
fatal: your current branch 'main' does not have any commits yet

Administrator@PC-20220115W0JX MINGW64 ~/repo/mywork (main)
```



把 gitfig 复制到根目录下

```
Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ cp ~/.gitconfig ./

Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ ls -l
total 0

Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ ls -al
total 5
drwxr-xr-x 1 Administrator 197121 0 Mar 16 23:48 ./
drwxr-xr-x 1 Administrator 197121 0 Mar 16 23:40 ../
drwxr-xr-x 1 Administrator 197121 0 Mar 16 23:31 .git/
-rw-r--r-- 1 Administrator 197121 193 Mar 16 23:48 .gitconfig

Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ !
```

git add .

```
Administrator@PC-20220115WOJX 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)
        .gitconfig

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

Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ git add .
warning: in the working copy of '.gitconfig', LF will be replaced by CRLF the
xt time Git touches it

Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ !
```

\$ git commit -m 'added some files for test'

```
Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ git commit -m 'added some files for test'
[main (root-commit) 1d04001] added some files for test
 1 file changed, 6 insertions(+)
 create mode 100644 .gitconfig

Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ git config --global user.name "TTATLib"

Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ git config --global user.email "TTATLib@noreply.gitcode.com"

Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ !
```

## 2. 输出结果解析

- `[main (root-commit) 1d04001] added some files for test:`
  - `main`: 当前提交所在的分支名。
  - `(root-commit)`: 表示这是仓库的 **首次提交** (仓库此前无提交记录, 此次提交是根提交)。
  - `1d04001`: 本次提交的唯一哈希值 (版本标识符)。
  - `added some files for test`: 用户写入的提交说明。
- `1 file changed, 6 insertions(+):`
  - 本次提交中, **1 个文件发生变更**, 新增了 **6 行内容** (`insertions(+)` 表示插入 / 新增代码行)。
- `create mode 100644 .gitconfig:`
  - 本次提交中 **新建了一个文件**, 文件名为 `.gitconfig`。
  - `100644`: 文件权限模式 (100 表示普通文件, 6 表示文件权限为 `rw-r--r--`, 即所有者可读可写, 其他用户只读)。

```
Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ cat ~/.gitconfig
[core]
    editor = "\"D:\\Documents\\首经贸\\作业\\金融编程与计算\\Microsoft VS Code\\bin\\code\" --wait
    quotepath = false
[user]
    name = TTATLib
    email = TTATLib@noreply.gitcode.com

Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ !
```

git log 查看提交

```
Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
$ git log
commit 1d040016b119dbcb70509576a92db518f13ab6b6 (HEAD -> main)
Author: TTATLib <TTATLib@noreply.gitcode.com>
Date:   Mon Mar 17 00:00:39 2025 +0800

    added some files for test

Administrator@PC-20220115WOJX MINGW64 ~/repo/mywork (main)
```

git push 提交到仓库

```
Administrator@PC-20220115W0JX MINGW64 ~/repo/mywork (main)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 408 bytes | 204.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Start Git Hooks Checking
To gitcode.com:TTATLib/mywork.git
 * [new branch]      main -> main
```