

## 第 2 周 命令行与文件系统 (初级)

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

```

tx58@tttx MINGW64 ~
$ ls
AppData
'Application Data'
Contacts
Cookies
Desktop
Documents
Downloads
Favorites
IntelGraphicsProfiles
Links
'Local Settings'
Music
'My Documents'
NTUSER.DAT
NTUSER.DAT{25371c46-6f67-11ed-9d53-e78acf4b81ca}.TM.blf
NTUSER.DAT{25371c46-6f67-11ed-9d53-e78acf4b81ca}.TMContainer000000000000000001.regtrans-ms
NTUSER.DAT{25371c46-6f67-11ed-9d53-e78acf4b81ca}.TMContainer000000000000000002.regtrans-ms
NetHood
Nox_share
OneDrive
Pictures
PrintHood
Recent
'Saved Games'
Searches
SendTo
Templates

```

```

0' '$' \225' '00' '$' \255' '00' '$' \210' '00' '$' \233' '0'
校' '$' \255' '00' '$' \224\237' '00' '$' \232' '工' 00' '$' \234'
00' '$' \210' '00' '$' \213' '0' '$' \233' '00' '$' \216' '000' '$' \200' '读稿件 .pdf'
0' '$' \211\207' '0' '$' \211\207' ' ' .docx'
0' '$' \224\237' '涯' 0' '$' \217\221' '00' '$' \225' '0' '$' \203' '0'
0' '$' \231' '0度' 00' '$' \221' '0' '$' \233\230' ' ' .lnk'
0' '$' \231' '0度' 翻' 00' '$' \221' ' ' .lnk'
稿' 0' '$' \255\220' '00' '$' \210' '00' '$' \213' '0' '$' \222\214' '0' '$' \200' '000' '$' \223' '0' '$' \203' '00' '$' \210\206' ' ' .doc
索' '$' \235\220'
0' '$' \213' '000' '$' \255' '0' '$' \210' '00' '$' \233' '0'
0' '$' \231\216' '0' '$' \211\231' '0' '$' \233' '00' '$' \222\255' ' ' .lnk'
00' '$' \206' '00' '$' \221'
00' '$' \205' '0' '$' \233' '0' .lnk'
00' '$' \231' '00' '$' \233' '0' '$' \226\207' '0' '$' \255\227' '00' '$' \214' '0' '$' \210\221' '0' '$' \234\213' '00' '$' \215' '0' '$' \201' ' ' (1) .pptx'
0' '$' \207\215' '00' '$' \201' '0' '$' \226\207' ' ' 件'
0' '$' \207\221' '0' '$' \236\215' '0' '$' \210' '00' '$' \233' '0'
0' '$' \207\221' '0' '$' \236\215' ' ' 计' 00' '$' \227' '0' '$' \234' '000' '$' \234' '00' '$' \232' 'week1.docx'
0' '$' \230\237' '徽'
0' '$' \231' '0天' 祥' 0' '$' \232\204' '工' 00' '$' \234' '0' '$' \233' '000' '$' \225' ' ' .lnk'
00' '$' \230' '00' '$' \223'

```

注意保存 ctrl+s



# ls(1) — Linux manual page

[NAME](#) | [SYNOPSIS](#) | [DESCRIPTION](#) | [AUTHOR](#) | [REPORTING BUGS](#) | [COPYRIGHT](#) | [SEE ALSO](#) | [COLOPHON](#)

Search online pages

LS(1)

User Commands

LS(1)

**NAME** [top](#)

ls - list directory contents

**SYNOPSIS** [top](#)

ls [*OPTION*]... [*FILE*]...

**DESCRIPTION** [top](#)

List information about the FILEs (the current directory by default). Sort entries alphabetically if none of `-cftuvSUX` nor `--sort` is specified.

Mandatory arguments to long options are mandatory for short options too.

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

`-A, --almost-all`  
do not list implied `.` and `..`

`--author`

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



```
ttx58@ttx MINGW64 ~/Desktop
$ cat 123.txt
123321 123456
ttx58@ttx MINGW64 ~/Desktop
$ cp 123.txt ~/Desktop/
cp: '123.txt' and '/c/Users/ttx58/Desktop/123.txt' are the same file

ttx58@ttx MINGW64 ~/Desktop
$ cp 123.txt ~/Downloads/

ttx58@ttx MINGW64 ~/Desktop
$ cp 123.txt ~/Downloads/qwe.txt

ttx58@ttx MINGW64 ~/Desktop
$
```

今天

 123.txt	2025/5/13 12:10	文本文档	1 KB
 qwe.txt	2025/5/13 12:11	文本文档	1 KB

```
ttx58@ttx MINGW64 ~/Desktop
$ ls -l fcl
total 1
-rw-r--r-- 1 ttx58 197609 13 May 13 11:11 123.txt

ttx58@ttx MINGW64 ~/Desktop
$ cp -r fcl ../Downloads/
```

今天

qwe.txt	2025/5/13 12:11	文本文档	1 KB
123.txt	2025/5/13 12:10	文本文档	1 KB
fcl	2025/5/13 12:17	文件夹	
fcl2	2025/5/13 12:21	文件夹	

- 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

```
ttx58@ttx MINGW64 ~/Desktop
$ ls -alh myproject/
total 34K
drwxr-xr-x 1 ttx58 197609  0 May 13 12:25 .
drwxr-xr-x 1 ttx58 197609  0 May 13 12:23 ..
-rw-r--r-- 1 ttx58 197609 13 May 13 12:10 123.txt
drwxr-xr-x 1 ttx58 197609  0 May 13 12:25 fcl
drwxr-xr-x 1 ttx58 197609  0 May 13 12:25 fcl2
-rw-r--r-- 1 ttx58 197609 13 May 13 12:11 qwe.txt

ttx58@ttx MINGW64 ~/Desktop
$
```

注意文件夹要加 `-r`

- 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

```
ttx58@ttx MINGW64 ~/Desktop
$ df
Filesystem            1K-blocks      Used Available Use% Mounted on
C:/Program Files/Git 209715196 166279408  43435788  80% /
D:                    289084412 172462600 116621812  60% /d

ttx58@ttx MINGW64 ~/Desktop
$ df -h
Filesystem            Size  Used Avail Use% Mounted on
C:/Program Files/Git 200G  159G   42G  80% /
D:                   276G  165G  112G  60% /d

ttx58@ttx MINGW64 ~/Desktop
$ ls -lh
total 70M
```

- 使用某个 AI 大模型（比如豆包，或 DeepSeek）解释下面的 Bash 命令  
这个 Bash 命令组合主要用于计算当前目录下每个文件或文件夹的磁盘使用情况，按从大到小排序，并将结果输出保存到 `~/report.txt` 文件中。下面是对命令各个部分的详细解释：

1. `**du -s *`**`:

- ``du`` 是 "disk usage" 的缩写，用来展示文件和目录的磁盘使用情况。

- ``-s`` 选项 (summarize) 指示 ``du`` 只显示每个参数的总计大小，而不是递归地列出所有子目录的大小。

- `*` 是通配符，表示当前目录下的所有文件和文件夹。因此，该命令会计算当前目录下每个文件和文件夹的总大小。

2. `**|`**``:

- 管道符 (`|``)，用于将前一个命令的输出作为后一个命令的输入。这里它将 ``du -s *` 的输出传递给 ``sort`` 命令。

3. `**`sort -nr`**`:

- ``sort`` 命令用于对文本行进行排序。

- ``-n`` 选项指定按照数值大小排序（而非默认的字典顺序），这对于正确排序如磁盘使用这样的数值数据是必要的。

- ``-r`` 选项指定以逆序（降序）方式排列结果，这样最大的数值将会排在最前面。

4. `**`> ~/report.txt`**`:

- 重定向操作符 (`>``) 用于将命令的输出重定向到一个文件中。如果文件已存在，则其内容会被覆盖；如果文件不存在，则会创建新文件。

- ``~/report.txt`` 表示用户主目录下的 ``report.txt`` 文件，命令执行的结果（即经过排序后的磁盘使用情况列表）将被写入此文件中。

总结起来，这条命令的作用是：计算当前目录下每个文件和目录的磁盘使用量，按从大到小的顺序对这些值进行排序，并将排序后的结果保存到主目录下的 ``report.txt`` 文件中。这可以用于快速查看哪些文件或目录占用了最多的磁盘空间。

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

```
ttx58@ttx MINGW64 ~
$ mkdir repo1

ttx58@ttx MINGW64 ~
$ cd repo1

ttx58@ttx MINGW64 ~/repo1
$ pwd
/c/Users/ttx58/repo1

ttx58@ttx MINGW64 ~/repo1
$ git clone git@gitcode.com:ttx581063/mywork.git
Cloning into 'mywork'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.

ttx58@ttx MINGW64 ~/repo1
$ ls
mywork

ttx58@ttx MINGW64 ~/repo1
$ ls -l
total 0
drwxr-xr-x 1 ttx58 197609 0 May 13 17:00 mywork
```

. 表示当前目录。

./ 是访问当前目录的一种方式（等价于当前目录的路径）。

所以 ./ 实际上就是告诉系统：“请把我复制到我当前所处的位置”。

```
ttx58@ttx MINGW64 ~/repo1/mywork (main)
$ git add .

ttx58@ttx MINGW64 ~/repo1/mywork (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   qwe.txt

ttx58@ttx MINGW64 ~/repo1/mywork (main)
$ git commit -m 'added some files for me'
[main c641886] added some files for me
 1 file changed, 1 insertion(+)
 create mode 100644 qwe.txt

ttx58@ttx MINGW64 ~/repo1/mywork (main)
$ cat ~/.gitconfig
[core]
    editor = \"D:\\Vscode\\Microsoft VS Code\\bin\\code\" --wait
    quotepath = false
[user]
    name = ttx581063
    email = ttx581063@noreply.gitcode.com
```

7. 去 第 02 周打卡 仓库，按照「操作步骤」说明，用 Fork-PR 方法提交本周学习报告