

任务目标

我们学习编程是为了 **自动化**，而图形界面是很难做到自动化的。所以在学习编程之前，初学者很有必要了解一些计算机的 **基础知识**，即图形界面下面的东西：

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)

目录树：是一种层次化的数据结构，用于组织和管理文件系统中的文件和目录。它形似倒置的树，根目录位于顶端，分支为子目录和文件。

根目录：目录树的最顶层节点，是所有文件和目录的起点。

路劲：指向文件或目录位置的字符串，描述其在目录树中的位置。

绝对路劲：从根目录开始的完整路径，唯一确定文件位置。

相对路劲：相对于当前工作目录的路径，依赖执行环境。

- 理解“当前工作目录”的概念，掌握 pwd 命令

当前工作目录是用户或程序在文件系统中“当前位置”的目录。所有相对直接输入 pwd

Pwd 基本用法：

\$ pwd

/home/user/projects # 输出示例 (Linux/macOS) 路径的操作（如访问文件、切换目录）都默认基于此目录。

- 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：简洁统一，以 `/` 为根，大小写敏感。

Windows 路径：依赖盘符，传统用 `\`，大小写不敏感。

最佳实践：

跨平台代码中优先使用正斜杠 `/`。

-避免硬编码路径，改用环境变量或相对路径。

-处理特殊字符（如空格）时用引号包裹路径。

掌握这些差异可显著减少跨平台开发中的路径相关问题，提升代码可移植性。

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

总结:

空格分隔: 命令、选项、参数必须用空格分隔。

短选项与长选项: 灵活使用 `-` 和 `--`, 注意合并规则。

参数处理: 区分选项参数与命令参数, 特殊字符需引号包裹。

跨平台一致性: Bash 与 Zsh 语法基本一致, 注意个别命令差异。

掌握这些规则后, 可通过 `man 命令名` (如 `man ls`) 查阅具体用法, 深入理解每个命令的选项和参数设计。

- 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

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/文件夹, 需 `-r` 处理目录。

mv: 移动或重命名文件/文件夹, 无需递归选项。

覆盖风险: 默认静默覆盖, 建议重要操作前用 `-i` 或备份数据。

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

以下是使用 `mkdir`、`ls` 和 `rm` 命令完成任务的详细步骤:

1. 创建 `myproject` 文件夹

使用 `mkdir` 命令创建名为 `myproject` 的文件夹:

```
``bash
```

```
mkdir myproject
```

```
...
```

2. 从图形界面复制文件/文件夹

Windows: 打开“文件资源管理器”, 选择文件/文件夹, 按 `Ctrl+C` 复制, 然后进入 `myproject` 文件夹, 按 `Ctrl+V` 粘贴。

macOS: 打开“访达”, 选择文件/文件夹, 按 `Cmd+C` 复制, 然后进入 `myproject` 文件夹, 按 `Cmd+V` 粘贴。

3. 使用 `ls` 查看文件/文件夹信息

进入 `myproject` 文件夹，使用 `ls` 查看文件/文件夹的大小和修改时间：

```
``bash
```

```
cd myproject
```

```
ls -lh
```

...

输出示例

```
``bash
```

```
rw-r--r--  1 user  staff   1.2K Mar 18 10:00 file1.txt
```

```
drwxr-xr-x  3 user  staff   96B Mar 18 10:01 folder1
```

...

`l`：显示详细信息（权限、大小、修改时间等）。

`h`：以人类可读格式显示文件大小（如 `1.2K`、`96B`）。

4. 使用 `rm` 删除文件/文件夹

删除文件

```
``bash
```

```
rm file1.txt
```

...

删除文件夹

使用 `-r` 选项递归删除文件夹及其内容：

```
``bash
```

```
rm -r folder1
```

...

强制删除（不提示确认）

添加 `-f` 选项：

```
``bash
```

```
rm -rf folder1
```

```
``
```

5. 完整流程示例

```
``bash
```

1. 创建 myproject 文件夹

```
mkdir myproject
```

2. 从图形界面复制文件/文件夹到 myproject

3. 进入 myproject 文件夹并查看内容

```
cd myproject
```

```
ls -lh
```

4. 删除文件/文件夹

```
rm file1.txt
```

```
rm -r folder1
```

```
``
```

注意事项

1. ****删除前确认****：`rm` 不可逆，建议先用 `ls` 确认目标。

2. ****递归删除****：删除文件夹必须加 `-r` 选项。

3. ****权限问题****：若权限不足，使用 `sudo`：

```
``bash
```

```
sudo rm -r folder1
```

```
``
```

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

以下是使用 `df` 和 `du` 命令查看磁盘空间的详细步骤及示例：

1. 使用 `df` 查看磁盘剩余空间

命令功能：显示文件系统的磁盘使用情况，包括总空间、已用空间、剩余空间及挂载点。

基本用法

```
```bash
```

```
df
```

```
```
```

输出示例：

```
```
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda1	20961280	12345678	7565432	62%	/
tmpfs	817596	0	817596	0%	/dev/shm

```
```
```

常用选项

`-h`（人类可读格式）：以 `K`、`M`、`G` 为单位显示。

```
```bash
```

```
df -h
```

```
```
```

输出示例：

```
```
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	20G	12G	7.2G	62%	/
tmpfs	799M	0	799M	0%	/dev/shm

```
```
```

查看特定文件系统：指定挂载点或设备名。

```
``bash
df -h /home # 查看 /home 分区的使用情况
...

```

显示文件系统类型：`-T` 选项。

```
``bash
df -Th
...

```

2. 使用 `du` 查看文件/文件夹占用空间**

****命令功能****：统计目录或文件的磁盘使用量。

基本用法

```
``bash
du
...

```

输出示例（递归显示当前目录下所有子目录的块数）：

```
...
4096  ./Documents
8192  ./Downloads
12288 .
...

```

常用选项

```
`-h`:
``bash
du -h
...

```

输出示例：

```
...  
  
4.0K  ./Documents  
8.0K  ./Downloads  
12K   .  
...
```

-汇总总大小：`-s` 选项。

```
``bash  
  
du -sh /home/user # 查看 /home/user 目录的总大小  
...
```

输出示例：

```
...  
  
2.5G  /home/user  
...
```

- 查看指定目录的详细信息：

```
``bash  
  
du -h /var/log # 显示 /var/log 下各子目录的大小  
...
```

-限制递归深度：`--max-depth=N`。

```
``bash  
  
du -h --max-depth=1 /home # 仅显示 /home 下一级子目录的大小  
...
```

3. 组合使用示例

步骤 1: 查看磁盘整体使用情况

```
``bash
```

```
df -h
```

```
...
```

目标: 确认哪个分区空间不足 (如 `/home` 使用率超过 80%)。

步骤 2: 定位大文件/文件夹

```
``bash
```

```
cd /home
```

```
du -sh * | sort -rh | head -n 5
```

```
...
```

说明:

- `du -sh *`: 列出 `/home` 下所有子目录和文件的大小。

- `sort -rh`: 按人类可读的数字逆序排序。

- `head -n 5`: 显示前 5 个结果。

输出示例:

```
...
```

```
12G  user1
```

```
5.8G  backups
```

```
3.2G  downloads
```

```
1.4G  media
```

```
800M  documents
```

```
...
```

步骤 3: 清理不需要的文件

```
``bash
```

```
rm -rf /home/user1/large_file.tar.gz # 删除大文件
```

```
...
```


4. 注意事项

- `df` 与 `du` 的差异:

- `df` 统计文件系统的元数据，可能包含已删除但被进程占用的空间。
- `du` 统计实际文件占用空间，结果更精确。

- 权限问题：无权限访问目录时，使用 `sudo`:

```
``bash
sudo du -sh /var/log
...`
```

避免误删：删除文件前务必确认路径，尤其是使用 `rm -rf`。

总结

`df`：快速查看磁盘整体剩余空间，常用 `df -h`。

`du`：详细分析文件/文件夹占用空间，常用 `du -sh` 或 `du -h --max-depth=1`。

- 组合技巧：结合 `sort`、`head` 等命令快速定位大文件。

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
```

这个命令的含义是：

1. `du -s *`：`du` 是 "disk usage" 的缩写，用于显示文件和目录的磁盘使用情况。`-s` 选项表示只显示每个目录或文件的总大小，而不显示其子目录或文件的大小。`*` 表示当前目录下的所有文件和目录。
2. `sort -nr`：`sort` 是用于对文本行进行排序的命令。`-n` 选项表示按数值大小进行排序，`-r` 选项表示降序排列（从大到小）。
3. `> ~/report.txt`：将前面命令的输出重定向到用户主目录下的 `report.txt` 文件中。如果文件不存在，则创建它；如果文件已存在，则覆盖其内容。

综合起来，这个命令的作用是：计算当前目录下所有文件和目录的磁盘使用情况，并按大小降序排列，将结果保存到用户主目录下的 `report.txt` 文件中。

修改后的命令

如果你想查看更详细的信息（例如显示每个文件和目录的大小，而不仅仅是总大小），可以去掉 `-s` 选项：

```
bash
```

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

如果你想查看人类可读的大小（例如以 KB、MB、GB 为单位），可以添加 ``-h`` 选项：

```
bash
```

```
du -h * | sort -hr > ~/report.txt
```

注意：``-h`` 选项与 ``-n`` 选项不兼容，因此需要使用 ``-h`` 选项来排序。

在计算机上应用

1. 打开终端（Linux/macOS）或命令提示符/PowerShell（Windows）。
2. 导航到你想要分析的目录，例如：

```
bash
```

```
cd /path/to/your/directory
```

3. 运行命令：

```
bash
```

```
du -h * | sort -hr > ~/report.txt
```

4. 完成后，打开 `~/report.txt`` 文件查看结果：

```
bash
```

```
cat ~/report.txt
```

注意事项

- 如果你在 Windows 上使用 PowerShell，``du`` 命令可能不可用。你可以使用类似的命令，例如 ``Get-ChildItem | Sort-Object Length -Descending`` 来获取类似的结果。

- 如果目录中有大量文件或子目录，命令可能需要一些时间才能完成。

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

已创建