

# 命令行、Python 与图像处理实验报告

王杰 (23160001074)

2025 年 9 月 19 日

## 1 实验目的

- 掌握常见 Shell 工作流：管道/重定向、作业控制、别名与配置等。
- 掌握 Python 基础语法：变量、流程控制、容器、函数、文件 I/O、异常、参数解析与虚拟环境。
- 初步使用 Python 图像处理库（Pillow、OpenCV、scikit-image、imageio）完成简单图像任务。

## 2 实验环境

- 操作系统：Windows 11 (WSL2: Ubuntu 24.04)
- Shell: bash (支持 jobs/fg/bg)
- Python: Python 3.11+, pip, venv
- 第三方库: pillow、opencv-python、scikit-image、imageio、matplotlib
- L<sup>A</sup>T<sub>E</sub>X 编译器: XeLaTeX; 在线: Overleaf

## 3 练习内容

### 3.1 命令行环境

#### 1. 查看命令帮助与手册 (快速定位用法)

```
ls --help | head -n 10
man grep    # q 退出; /pattern 搜索; n 跳下一个
```

```
tingol@tingol-VMware-Virtual-Platform:~$ ls --help | head -n 10
用法: ls [选项]... [文件]...
列出 <文件> (默认为当前目录) 的信息。
如果既没有指定 -cftuvSUX 中任何一个，也没有指定 --sort，则按字母排序项目。

长选项的必选参数对于短选项也是必选的。
-a, --all          不要隐藏以 . 开头的项目
-A, --almost-all   列出除 . 及 .. 以外的所有项目
--author          与 -l 同时使用时，列出每个文件的作者
-b, --escape        以 C 风格的转义序列表示不可打印的字符
--block-size=大小    与 -l 同时使用时，打印大小前将其除以 <大小>;
tingol@tingol-VMware-Virtual-Platform:~$ man grep
```

## 2. 重定向与管道：去重计数 Top N

```
cat /var/log/syslog 2>/dev/null \
| awk '{print $5}' | sort | uniq -c | sort -nr | head
```

```
tingol@tingol-VMware-Virtual-Platform:~$ cat /var/log/syslog 2>/dev/null | awk '{print $5}' | sort | uniq -c | sort -nr
| head
41 Deactivated
28 sysstat-collect.service
16 NetworkManager-dispatcher.service
15 CMD
11 Successfully
11 GLib-GIO-WARNING
11 Activating
9 Bus
8 Got
7 type=1400
```

## 3. 通配符与花括号扩展：批量造数据

```
mkdir demo && cd demo
touch file-{a..c}{1..3}.txt
ls file-a*.txt
```

```
tingol@tingol-VMware-Virtual-Platform:~/25q2$ mkdir -p demo
tingol@tingol-VMware-Virtual-Platform:~/25q2$ cd demo
tingol@tingol-VMware-Virtual-Platform:~/25q2/demo$ touch file-{a..c}{1..3}.txt
tingol@tingol-VMware-Virtual-Platform:~/25q2/demo$ ls file-a*.txt
file-a1.txt file-a2.txt file-a3.txt
tingol@tingol-VMware-Virtual-Platform:~/25q2/demo$ cd ..
```

## 4. 命令替换与 shell

```
echo "Today is $(date +%F). Files here: $(ls | wc -l)."
( cd /etc && echo "In /etc: $(pwd)" )
```

```
tingol@tingol-VMware-Virtual-Platform:~/25q2$ echo "Today is $(date +%F). Files here: $(ls | wc -l)."
Today is 2025-09-19. Files here: 1.
tingol@tingol-VMware-Virtual-Platform:~/25q2$ (
> cd /etc
> echo "In /etc: $(pwd)"
> )
In /etc: /etc
```

## 5. 环境变量与别名配置

```
echo $PATH
export EDITOR=vim
alias ll='ls -alh'      # 将别名写入 ~/.bashrc 可持久化
```

```
tingol@tingol-VMware-Virtual-Platform:~/25q2$ echo "$PATH"
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/usr/games:/usr/local/games:/snap/bin
tingol@tingol-VMware-Virtual-Platform:~/25q2$ export EDITOR=vim
tingol@tingol-VMware-Virtual-Platform:~/25q2$ alias ll='ls -alh'
总计 12K
drwxrwxr-x  3 tingol tingol 4.0K  9月 19 22:42 .
drwxr-x--- 16 tingol tingol 4.0K  9月 19 22:42 ..
drwxrwxr-x  2 tingol tingol 4.0K  9月 19 22:43 demo
```

## 6. 作业控制与后台运行 (Ctrl+Z 暂停, bg/fg 切换)

```
sleep 1000 &
jobs          # 查看作业
fg %1         # 回到前台
sleep 2000 & disown # 让进程脱离终端
```

```
tingol@tingol-VMware-Virtual-Platform:~/25q2$ sleep 300 &
[1] 18041
tingol@tingol-VMware-Virtual-Platform:~/25q2$ pgrep -af "sleep 300"
18041 sleep 300
```

## 7. 查找并结束进程 (更优雅的方式)

```
pgrep -af sleep      # 查 PID
pkill -f sleep       # 按名称结束, 不手敲 PID
```

```
tingol@tingol-VMware-Virtual-Platform:~/25q2$ sleep 200 &
[2] 18075
tingol@tingol-VMware-Virtual-Platform:~/25q2$ pgrep -af "sleep 200"
18075 sleep 200
tingol@tingol-VMware-Virtual-Platform:~/25q2$ pkill -f "sleep 200"
[2]+  已终止                  sleep 200
```

## 3.2 Python 基础

### 1. Hello, World 与 shebang

```
#!/usr/bin/env python3
print("Hello, World")
```

### 2. 变量与类型、f-string

```
name = "Alice"; score = 92.5
print(f"{name}'s score is {score:.1f}")
```

### 3. 条件与循环: 奇数求和

```
total = 0
for i in range(1, 100):
    if i % 2 == 1:
        total += i
print(total)
```

#### 4. 列表/字典推导式

```
squares = [x*x for x in range(6)]
freq = {ch: "hello world".count(ch) for ch in set("hello world")}
print(squares); print(freq)
```

#### 5. 函数与类型注解、文档字符串

```
def area(w: float, h: float) -> float:
    """Return rectangle area = w*h."""
    return w * h

print(area(3.0, 4.5))
```

#### 6. 文件 I/O (UTF-8)

```
from pathlib import Path
Path("note.txt").write_text("第一行\n第二行\n", encoding="utf-8")
print(Path("note.txt").read_text(encoding="utf-8"))
```

#### 7. 异常处理 (try/except/else/finally)

```
try:
    x = int(input("Enter an integer: "))
except ValueError as e:
    print("Invalid!", e)
else:
    print("x^2 =", x*x)
finally:
    print("Done.")
```

#### 8. 命令行参数解析 (argparse)

```
# save as add.py: python add.py 3 5 -> 8
import argparse
p = argparse.ArgumentParser()
p.add_argument("a", type=int); p.add_argument("b", type=int)
args = p.parse_args()
print(args.a + args.b)
```

#### 9. 虚拟环境与依赖导出

```
python -m venv .venv
source .venv/bin/activate # Windows: .venv\Scripts\activate
pip install requests
pip freeze > requirements.txt
```

### 3.3 图像处理

#### 1. Pillow: 打开/缩放/灰度化/保存

```
from PIL import Image
im = Image.open("input.jpg")
im2 = im.resize((512, 512)).convert("L")
im2.save("output_pillow_gray.jpg")
```

#### 2. OpenCV: 高斯模糊 + Canny 边缘

```
import cv2
img = cv2.imread("input.jpg")
blur = cv2.GaussianBlur(img, (5,5), 0)
edge = cv2.Canny(blur, 100, 200)
cv2.imwrite("output_cv_edge.png", edge)
```

#### 3. scikit-image: Otsu 二值化

```
from skimage import io, filters, img_as_ubyte
img = io.imread("input.jpg", as_gray=True)
th = filters.threshold_otsu(img)
binary = img > th
io.imsave("output_skimage_binary.png", img_as_ubyte(binary))
```

#### 4. imageio: 合成 GIF (配合 matplotlib 预览)

```
import imageio.v2 as imageio
import glob
frames = [imageio.imread(p) for p in sorted(glob.glob("frames/*.png"))]
imageio.mimsave("anim.gif", frames, duration=0.08)
print("Saved anim.gif")
```

## 4 常见问题与解决

- **命令行作业控制失效**: 确认当前 Shell 为 bash/zsh，并使用 jobs/fg/bg；脱离终端用 disown 或 nohup。
- **Unicode 编码问题**: 统一使用 UTF-8；文件读写指定 encoding="utf-8"。

## 5 心得体会

通过 Missing Semester 的作业控制、别名与配置，命令行工作流显著顺滑；Python 基础结合 argparse/pathlib 等现代用法更易维护；图像处理库各擅胜场——Pillow 上手最快、OpenCV 算法丰富、scikit-image API 语义清晰、imageio 用于读写动画便捷。将三者结合，可以快速完成“命令行调度—Python 处理—结果产出”的闭环。

## 6 参考资料

- 命令行环境(中文):<https://missing-semester-cn.github.io/2020/command-line/>
- Command-line Environment (英文原文):<https://missing.csail.mit.edu/2020/command-line/>
- Python 基础教程/3.x 教程 (菜鸟教程): <https://www.runoob.com/python/python-tutorial.html>, <https://www.runoob.com/python3/python3-tutorial.html>
- CSDN: 【Python】推荐五个常用的图像处理库: <https://blog.csdn.net/sgzqc/article/details/124871774>

## 7 github 地址

<https://github.com/tingol666/git2025q2.git>