

# 系统开发工具基础实验报告 3

姓名：刘浩洋

学号：24040021022

班级：软件工程

实验日期：2025 年 9 月 12 日

## 一、实验目的

1. 掌握 Linux 命令行中任务与进程的管理方法，包括前台/后台切换、暂停、恢复与终止；
2. 熟悉 Python 基本语法，掌握变量、控制结构、函数和模块的使用；
3. 初步了解 OpenCV 库的基本功能，实现图像的读取、显示与简单处理；

## 二、实验环境

- 操作系统：Windows 11
- 虚拟化平台：VMware Workstation Pro 17
- 虚拟机系统：Ubuntu 24.04.3 LTS (64 位)
- Shell 环境：Bash
- 编程语言：Python 3
- 图像处理库：OpenCV-Python (cv2)

## 三、练习内容

本次实验主要包括：

- 使用 Shell 进行进程的启动、暂停、后台运行与信号控制；

- 编写 Python 脚本实现基本逻辑与函数调用；
- 使用 OpenCV 进行图像的加载、显示与颜色空间转换；
- 结合 count.py 和 fib.py 演示任务控制机制；

## 四、20 个实例（命令行、Python 与 OpenCV）

### 实例 1：创建实验目录结构

为组织本次实验文件，创建包含子目录的项目结构。使用 -p 参数可安全创建多级目录。

```
mkdir -p /home/lhy/lesson3/{scripts,images,docs}
cd /home/lhy/lesson3
touch scripts/count.py scripts/fib.py
ls -R
pwd
```

### 实例 2：编写 count.py（无限计数）

编写一个持续输出递增数字的 Python 脚本，用于演示长时间运行任务的进程控制。

```
#!/usr/bin/env python3
import time
print(" 开始计数")
number = 1
while True:
    print(number)
    number += 1
    time.sleep(2)
```

### 实例 3：编写 fib.py（斐波那契计算）

编写一个计算前 50 项斐波那契数列的脚本，用于演示耗时任务的后台运行与监控。

```
#!/usr/bin/env python3
import time
def fb(n):
    if n == 1 or n == 2:
        return 1
    return fb(n-1) + fb(n-2)
start = time.time()
for i in range(1, 51):
    time.sleep(2)
    print('fb(%d)=%d' % (i, fb(i)))
end = time.time()
print(' 总时间为: %fs' % (end - start))
```

### 实例 4：在前台运行 Python 脚本

直接执行脚本，它将在前台占用终端，期间无法输入其他命令，按 Ctrl+C 可中断。

```
python3 scripts/count.py
```

### 实例 5：暂停前台任务（Ctrl+Z）

在脚本运行时按 Ctrl+Z，可将任务暂停并送入后台，状态为 Stopped。

# 运行中按 Ctrl+Z

```
[1]+ Stopped python3 scripts/count.py
```

### 实例 6：后台运行脚本（&）

在命令末尾加 &，可让脚本在后台运行，终端立即返回，可继续执行其他命令。

```
python3 scripts/fib.py &
```

```
[1] 12345
```

# 脚本在后台运行，PID 为 12345

### 实例 7：查看后台任务（jobs）

使用 jobs 命令查看当前 Shell 会话中的后台任务，包括运行中和暂停的任务及其状态。

```
jobs
```

```
[1]+ Running python3 scripts/fib.py &
```

```
[2]- Stopped python3 scripts/count.py
```

#### 实例 8：恢复暂停任务到后台 (bg)

使用 bg 命令可将暂停的任务在后台继续运行。

```
bg %2  
[2]+ python3 scripts/count.py &
```

#### 实例 9：将后台任务调回前台 (fg)

使用 fg 命令可将后台任务（运行或暂停）调回前台继续交互。

```
fg %1  
# fib.py 恢复到前台，终端被占用
```

#### 实例 10：终止进程 (kill)

使用 kill 命令向进程发送信号。默认发送 SIGTERM，-9 为强制终止 SIGKILL。

```
kill %1  
kill -9 12345  
jobs  
# 查看是否已终止
```

#### 实例 11：Python 变量与数据类型

Python 支持动态类型，常用类型包括 int、float、str、bool。

```
#!/usr/bin/env python3  
name = " 刘浩洋"  
age = 20  
height = 1.75  
isstudent = True  
print(f" 姓名: {name}, 年龄: {age}")  
print(f" 身高: {heightm}, 学生: {isstudent}")
```

### 实例 12: Python 控制结构 (if-elif-else)

使用条件语句实现逻辑分支，注意缩进。

```
#!/usr/bin/env python3
score = 85
if score >= 90:
    print(" 优秀")
elif score >= 80:
    print(" 良好")
elif score >= 60:
    print(" 及格")
else:
    print(" 不及格")
```

### 实例 13: Python 循环 (for 与 while)

for 循环常用于遍历序列，while 用于条件循环。

```
#!/usr/bin/env python3
for 循环
fruits = [" 苹果", " 香蕉", " 橙子"]
for fruit in fruits:
    print(f" 我喜欢 {fruit}")

# while 循环
i = 0
while i < 3:
    print(f" 计数: {i}")
    i += 1
```

### 实例 14: Python 函数定义与调用

函数用于封装可复用代码，支持参数与返回值。

```
#!/usr/bin/env python3
def greet(name, lang=" 中文"):
    return f" 你好, {name!} 使用 {lang} 打招呼。"

print(greet(" 张三"))
print(greet(" 李四", "Python"))
```

### 实例 15: 导入 Python 模块

使用 import 导入标准库或第三方库。

```
#!/usr/bin/env python3
import math
import datetime

print(" 圆周率: ", math.pi)
print(" 当前时间: ", datetime.datetime.now())
```

### 实例 16: 安装 OpenCV 库

使用 pip 安装 OpenCV-Python 库, 需确保已安装 pip 和相关依赖。

```
sudo apt update
sudo apt install python3-pip
pip3 install opencv-python
```

### 实例 17: 读取并显示图像 (OpenCV)

使用 cv2.imread() 读取图像, cv2.imshow() 显示。

```
#!/usr/bin/env python3
import cv2

img = cv2.imread("images/test.jpg")
if img is not None:
    cv2.imshow(" 图像", img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
else:
    print(" 图像加载失败! ")
```

### 实例 18: 图像亮度调整

使用 `cv2.convertScaleAbs()` 调整图像亮度。通过 `input` 获取用户输入的亮度值 (-100 到 100)，负值变暗，正值变亮，处理后保存为 `pic.png`。

```
#!/usr/bin/env python3
import cv2
img = cv2.imread("images/test.jpg")
if img is None:
    print(" 错误: 图像加载失败, 请检查路径! ")
else:
    value = int(input(" 亮度值 (-100 到 100): "))
    img_adjusted = cv2.convertScaleAbs(img, beta=value)
    cv2.imwrite("pic.png", img_adjusted)
    print(" 亮度已调整并保存")
```

### 实例 19: 保存处理后的图像

使用 `cv2.imwrite()` 将处理结果保存到文件。

```
#!/usr/bin/env python3
import cv2
img = cv2.imread("images/test.jpg")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imwrite("images/gray.jpg", gray)
print(" 灰度图像已保存! ")
```

### 实例 20: 综合任务控制演示

综合使用 `count.py` 和 `fib.py` 演示完整任务控制流程。

```
# 1. 后台运行计数
python3 scripts/count.py &
# 2. 后台运行斐波那契
python3 scripts/fib.py &
# 3. 查看任务
jobs
# 4. 暂停其中一个
kill -STOP %1
# 5. 恢复
kill -CONT %1
# 6. 终止全部
kill %1 %2
```

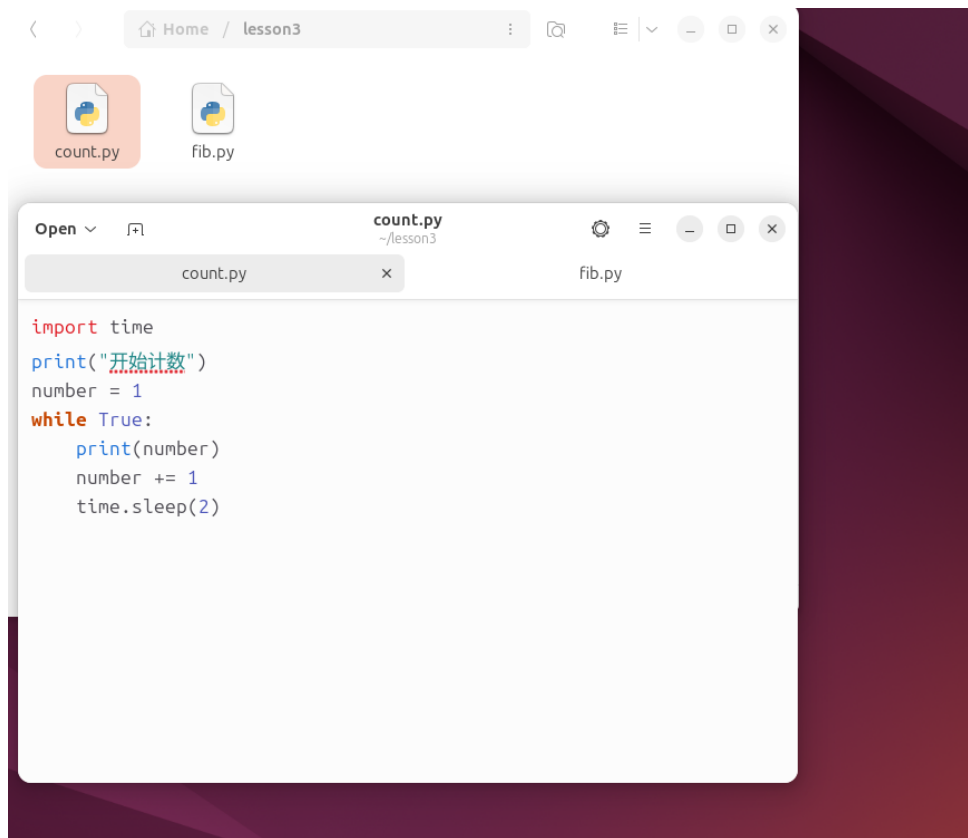


图 1: lec3-1



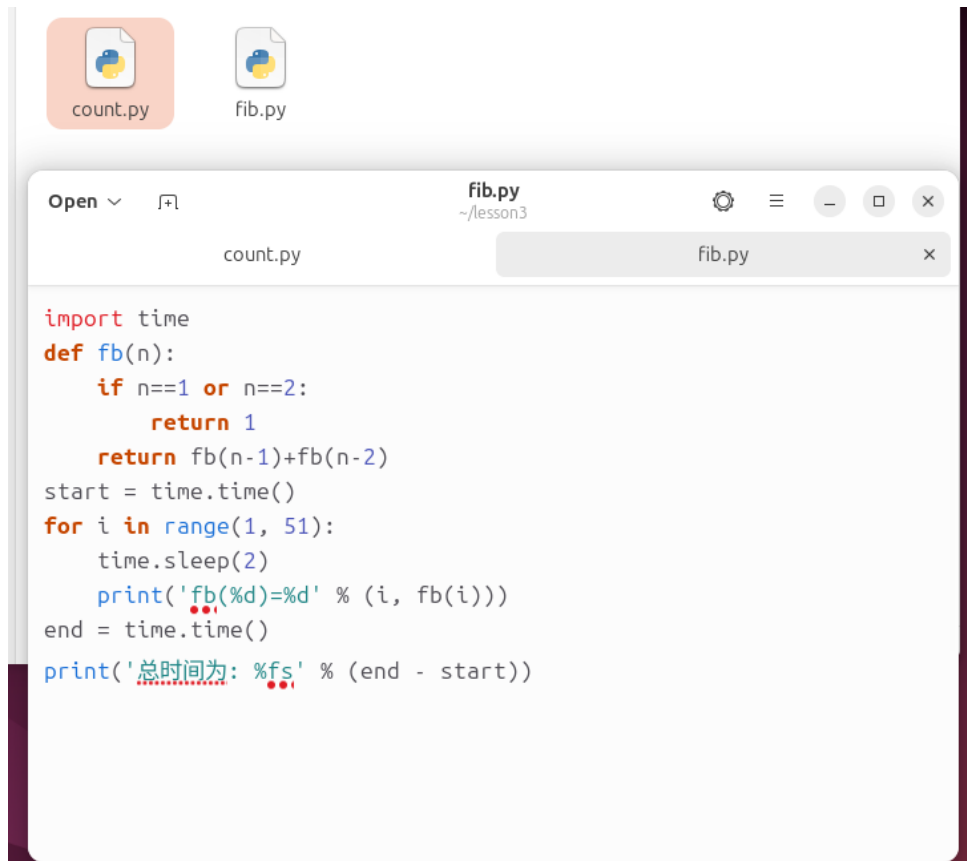


图 2: lec3-2

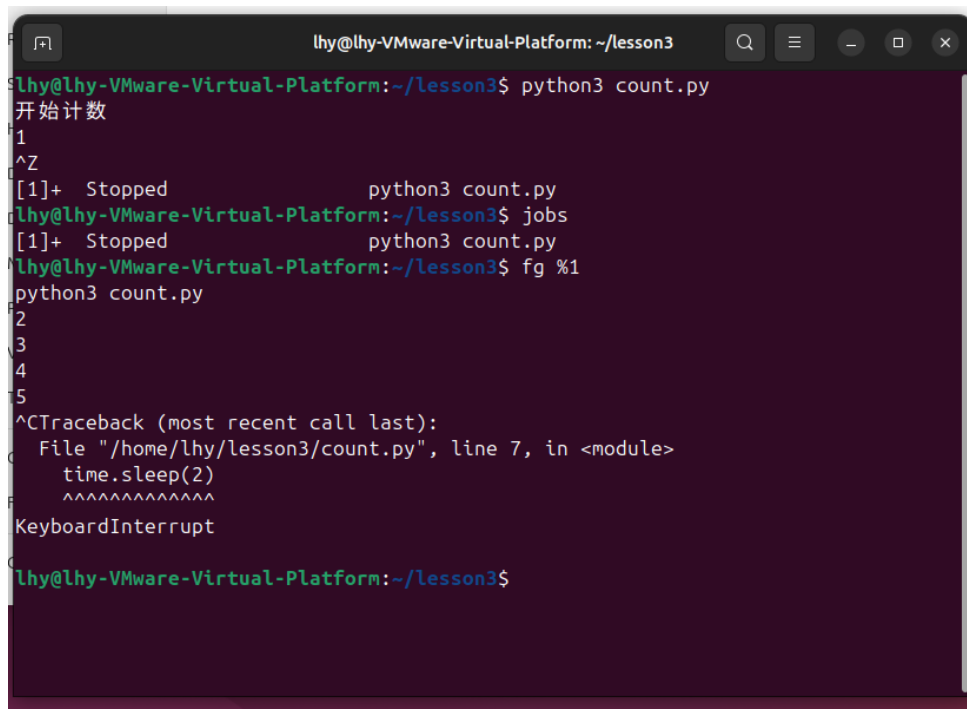


图 3: lec3-3

```
lhy@lhy-VMware-Virtual-Platform: ~/lesson3
lhy@lhy-VMware-Virtual-Platform:~/lesson3$ python3 fib.py &
[1] 4227
lhy@lhy-VMware-Virtual-Platform:~/lesson3$ fb(1)=1
fb(2)=1
kill -STOP fb(3)=2
%1
lhy@lhy-VMware-Virtual-Platform:~/lesson3$ jobs
[1]+  Stopped                  python3 fib.py
lhy@lhy-VMware-Virtual-Platform:~/lesson3$ bg %1
[1]+  python3 fib.py &
lhy@lhy-VMware-Virtual-Platform:~/lesson3$ fb(4)=3
fb(5)=5
fb(6)=8
kill %1
lhy@lhy-VMware-Virtual-Platform:~/lesson3$ jobs
[1]+  Terminated              python3 fib.py
lhy@lhy-VMware-Virtual-Platform:~/lesson3$ jobs
lhy@lhy-VMware-Virtual-Platform:~/lesson3$
```

图 4: lec3-4

```
Thonny - E:\Users\lenovo\Desktop\开发工具基础\lesson3\opencv\picture.py @ 50:22
文件 编辑 视图 运行 工具 帮助
picture.py
1 import cv2
2 import numpy as np
3
4 # 读取图像
5 img = cv2.imread("pic.png")
6 if img is None:
7     print("无法读取图像，请检查路径")
8     exit()
9
10 while True:
11     print("\n图像处理演示")
12     print("1. 调整亮度")
13     print("2. 调整对比度")
14     print("3. 缩放图像")
15     print("4. 显示图像")
16     print("0. 退出")
17
18     choice = input("请选择操作: ")
19
20     if choice == '0':
21         break
22
23     elif choice == '1':
24         value = int(input("亮度值(-100到100): "))
25         img = cv2.convertScaleAbs(img, beta=value)
26         cv2.imwrite("pic.png", img)
27         print("亮度已调整并保存")
28
29     elif choice == '2':
30         value = float(input("对比度值(0.5-3.0): "))
31         img = cv2.convertScaleAbs(img, alpha=value)
32         cv2.imwrite("pic.png", img)
33         print("对比度已调整并保存")
34
35     elif choice == '3':
36         ...
37
38 Shell
Python 3.10.11 (C:\Thonny\python.exe)
>>>
```

图 5: lec3-5

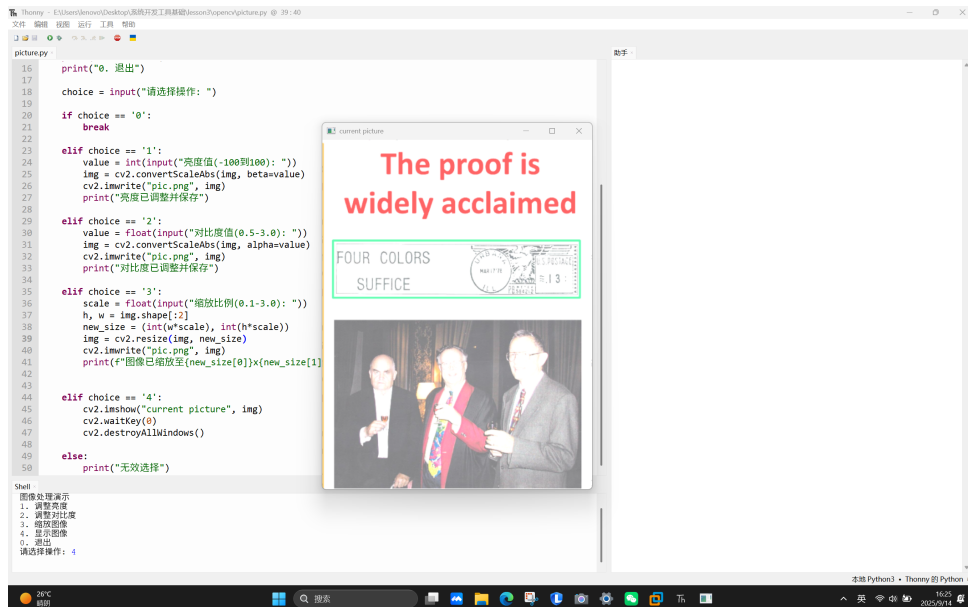


图 6: lec3-6

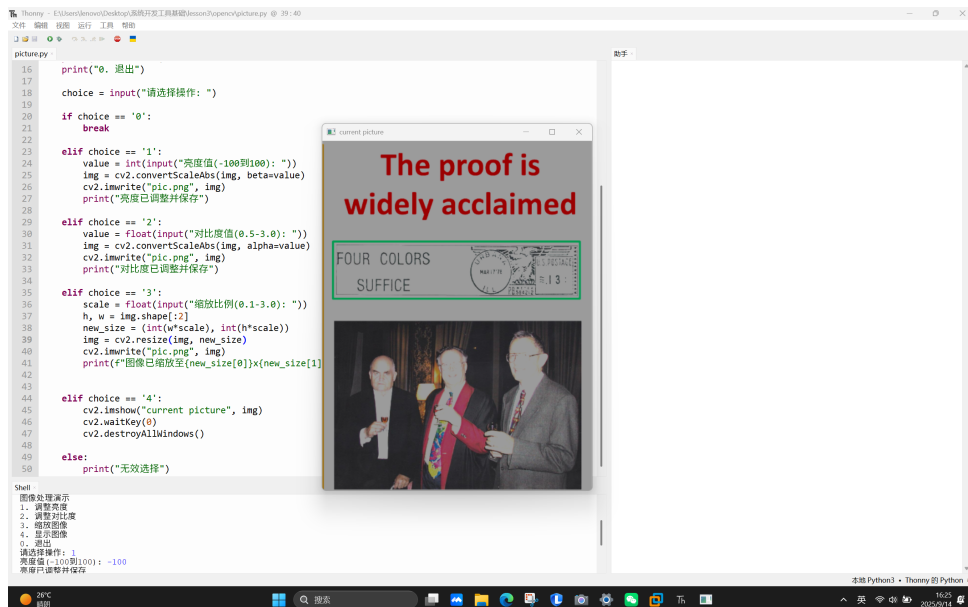


图 7: lec3-7

```
MINGW64/e/Users/lenovo/Desktop/系统开发工具基础
lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git add .
lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git commit -m "实验3"
[master 56e47cf] 实验3
12 files changed, 69 insertions(+)
create mode 100644 lesson3/count.py
create mode 100644 lesson3/fib.py
create mode 100644 lesson3/lec3 (1).png
create mode 100644 lesson3/lec3 (2).png
create mode 100644 lesson3/lec3 (3).png
create mode 100644 lesson3/lec3 (4).png
create mode 100644 lesson3/lec3 (5).png
create mode 100644 lesson3/lec3 (6).png
create mode 100644 lesson3/lec3 (7).png
create mode 100644 lesson3/opencv/pic.png
create mode 100644 lesson3/opencv/picture.py
create mode 100644 "lesson3/opencv/345\261\217\345\271\225\346\210\252\345\233\276 .png"
lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ git push origin master
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 18 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (16/16), 2.50 MiB | 1.46 MiB/s, done.
Total 16 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:ouc-lhy/for-lesson.git
   aff13c3..56e47cf master -> master
lenovo@LAPTOP-66EK36EJ MINGW64 /e/Users/lenovo/Desktop/系统开发工具基础 (master)
$ |
```

图 8: commit 截图

## 五、实验结果

- 成功实现对 Python 脚本的进程控制（前后台切换、暂停、终止）；
- 能够使用 jobs、fg、bg、kill 等命令管理任务；
- 掌握 Python 基本语法与函数定义；
- 成功安装并使用 OpenCV 进行图像读取、转换与保存；
- 所有脚本均能正确运行,实验代码已整理并提交至 <https://github.com/ouc-lhy/for-lesson/tree/master/lesson3>。

## 六、解题感悟

本次实验深入学习了 Linux 任务控制机制，理解了进程状态转换与信号处理的重要性。通过 count.py 和 fib.py 的实际操作，掌握了如何管理长时间运行的任务。Python 部分让我快速入门了这门强大的语言，其简洁语法和丰富库支持令人印象深刻。OpenCV 的引入打开了计算机视觉的大门，图像处理的直观效果增强了学习兴趣。整个实验过程锻炼了我整合 Shell、Python 与系统工具的能力，为后续开发打下了坚实基础。

## 七、GitHub 链接

<https://github.com/ouc-lhy/for-lesson/tree/master/lesson3>