# 系统开发工具基础实验报告

姓名: 李佳潼　学号: 22020007043　年级专业: 2022 级计算机科学与技术

实验名称: 命令行控制　实验时间: 2024 年 9 月 6 日

## 1 实验目的

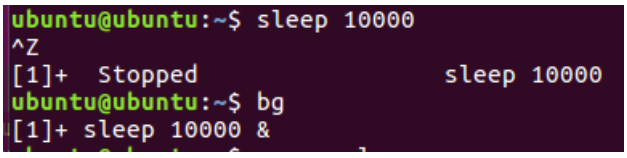
- 进一步学习命令行相关控制命令
- 学习 python 基础语法
- 了解并学习任务控制、SSH 远端设备控制、python 逻辑处理、图像处理等内容
- 实现相关常见现实问题的解决方法

## 2 实验内容及结果

### 2.1 课堂练习部分-命令行环境

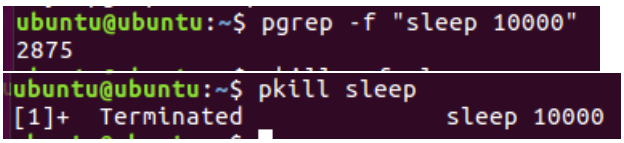
1. 在终端中执行 sleep 10000 这个任务。然后用 Ctrl-Z 将其切换到后台并使用 bg 来继续允许它

```
sleep 10000
bg
```




2. 使用 pgrep 来查找 pid 并使用 pkill 结束进程而不需要手动输入 pid

```
pgrep -f "sleep 10000"
pkill sleep
```



3. 请编写一个 bash 函数 pidwait ，它接受一个 pid 作为输入参数，然后一直等待直到该进程结束。您需要使用 sleep 来避免浪费 CPU 性能

   编写函数如图

```
ubuntu@ubuntu:~$ pidwait()
> {
> while kill -0 $1
> do
> sleep 1
> done
> ls
> }
```

   运行 pidwait，如图

```
ubuntu@ubuntu:~$ sleep 60 & pidwait $(pgrep -f "sleep 60")
[2] 3469
[1]-  Done                    sleep 60
bash: kill: (3463) - No such process
all.sh     Documents         html_root          Music
all.txt    Downloads         ICS                occurance.txt
```

4. 创建一个 dc 别名，它的功能是当我们错误的将 cd 输入为 dc 时也能正确执行

```
alias dc=cd
```

```
ubuntu@ubuntu:~$ cd history
ubuntu@ubuntu:~/history$ cd ..
ubuntu@ubuntu:~$ alias dc=cd
ubuntu@ubuntu:~$ dc history
ubuntu@ubuntu:~/history$ dc ..
ubuntu@ubuntu:~$
```

   可以看到 dc 已经被赋予同 cd 相同的指令功能

5. 获取您最常用的十条命令，尝试为它们创建别名

```
history | awk '{$1="";print substr($0,2)}' | sort | uniq -c |
   sort -n | tail -n 10
```

```
ubuntu@ubuntu:~$ history | awk '{$1="";print substr($0,2)}' | sort | uniq -c | s
ort -n | tail -n 10
      9 ls
     10 cd ../
     11 gdb -q bufbomb
     12 cd ..
     14 ./bomb
     19 ./bomb ans.txt
     24 cd bomb37
     24 ./driver.py
     28 gdb bomb
     32 make
```

6. 获取您使用频率最少的十条命令，尝试为它们创建别名

```
history | awk '{$1="";print substr($0,2)}' | sort | uniq -c |
   sort -rn | tail -n 10
```

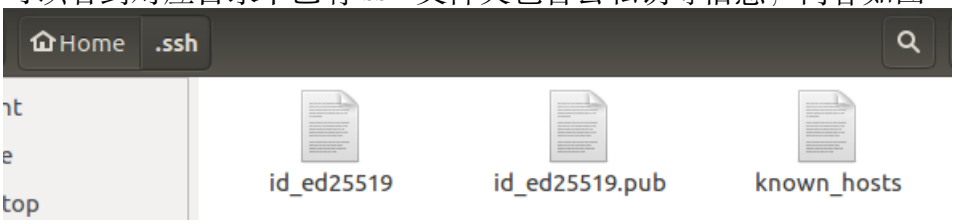
```
ubuntu@ubuntu:~$ history | awk '{$1="";print substr($0,2)}' | sort | uniq -c | s
ort -rn | tail -n 10
      1 cat test2.txt
      1 cat /proc/sys/kernel/randomize_va_space
      1 cat out1.log | grep Everything | wc -l
      1 cat Level2.txt | ./hex2raw | ./bufbomb -u 22020007043
      1 catkin_make
      1 cat /etc/resolv.conf
      1 .bomb ans.txt
      1 bg
      1 bash
      1 b *0x804934e
```

7. 创建 SSH 密钥对

```
ssh-keygen -o -a 100 -t ed25519
```

```
ubuntu@ubuntu:~$ ssh-keygen -o -a 100 -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519.
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:qr9gvjiOk4SoNNvTrhGR+1xin23Kq4rXcaK60GNxejY ubuntu@ubuntu
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|     .           |
|    o            |
|     o           |
|o .o.o .S        |
|o= +=o+oo        |
|=.O.Eo=o o       |
|=+o%.*. o        |
|.**+O=+=.        |
+----[SHA256]-----+
```

可以看到对应目录下已有.ssh 文件夹包含公私钥等信息，内容如图




## 2.2 课下练习部分-python 的基础运用

1. 进行基础的数学运算

```
a = 5
b = 3
sum_result = a + b
difference_result = a - b
```

```
product_result = a * b
quotient_result = a / b
print ( f"Sum: { sum_result }")
print ( f" Difference : { difference_result }")
print ( f" Product : { product_result }")
print ( f" Quotient : { quotient_result }")
```

运行结果如图

```
D:\anaconda\python.exe D:\pycharmProjects\pythonProject\1_add.py
Sum: 8
 Difference : 2
 Product : 15
 Quotient : 1.6666666666666667

进程已结束，退出代码为 0
```

分别输出加、减、乘、整除的结果，结果正确

2. 定义并调用一个简单函数

```
def greet (name ) :
    return f" Hello , {name}!"
name = input (" Enter your name : ")
print ( greet (name ) )
```

运行结果如图

```
D:\anaconda\python.exe D:\pycharmProjects\pythonProject\1_add.py
Sum: 8
 Difference : 2
 Product : 15
 Quotient : 1.6666666666666667

进程已结束，退出代码为 0
```

输入姓名实现带姓名的预期输出，函数调用成功

3. 创建和访问列表

```
fruits = [ " apple " , "banana " , " cherry " ]
print (fruits[0])
fruits . append (" date ")
print (fruits)
```

运行结果如图

```
D:\anaconda\python.exe D:\pycharmProjects\pythonProject\1_list.py
 apple
[' apple ', 'banana ', ' cherry ', ' date ']

进程已结束，退出代码为 0
```

返回列表数据，访问列表成功

4. 使用条件判断与循环

```python
# 条件判断
x = 15
if x > 10:
    print("x 大于 10")
else:
    print("x 小于或等于 10")

# for 循环
for i in range(5):
    print(f"循环次数: {i}")

# while 循环
n = 0
while n < 5:
    print(f"当前值: {n}")
    n += 1
```

运行结果如图

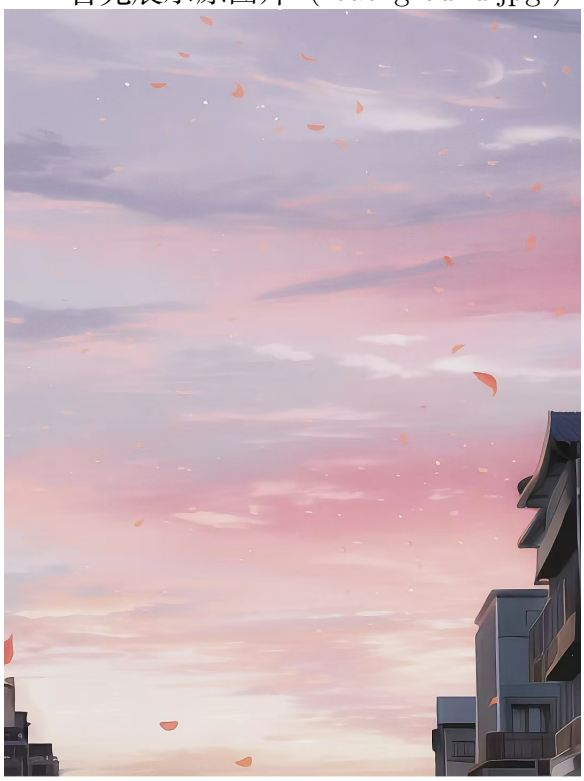
```
D:\anaconda\python.exe D:\pycharmProjects\pythonProject\xunhuan.py
x 大于 10
循环次数: 0
循环次数: 1
循环次数: 2
循环次数: 3
循环次数: 4
当前值: 0
当前值: 1
当前值: 2
当前值: 3
当前值: 4

进程已结束，退出代码为 0
```

输出每次条件判断和循环的结果，逻辑实现正确

## 2.3 课下练习部分-python 图片处理

首先展示原图片（'background.jpg'）



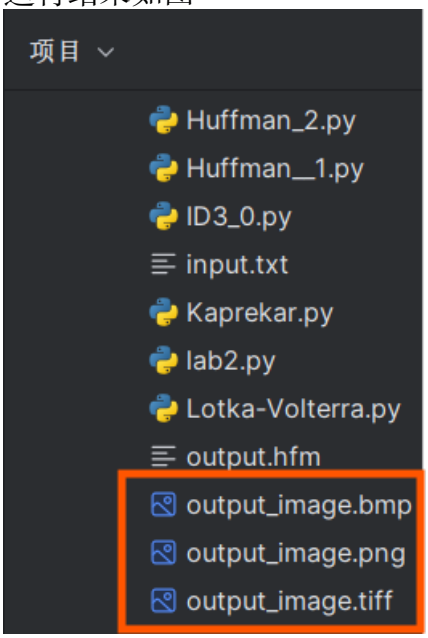
1. 转换图像格式

```python
from PIL import Image
def save_image(image_path, output_path, format='PNG'):
    image = Image.open('background.jpg')

    image.save(output_path, format=format)

save_image('background.jpg', 'output_image.png', format='PNG')
save_image('background.jpg', 'output_image.bmp', format='BMP')
save_image('background.jpg', 'output_image.tiff', format='TIFF')
```

运行结果如图




目录下分别新建.png、.bmp、.tiff 文件，实现格式转换

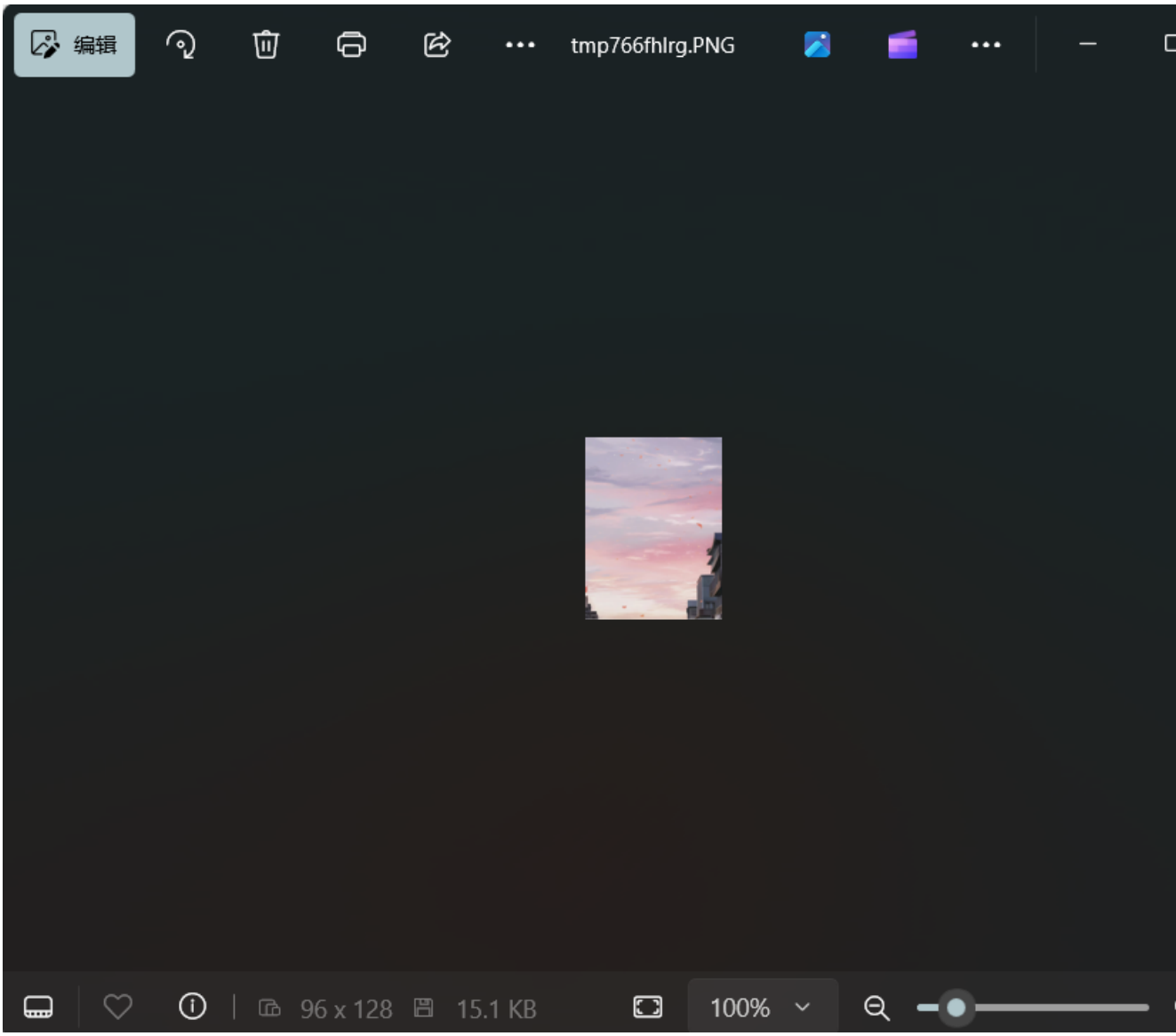
2. 创建缩略图

```python
from PIL import Image

im = Image.open('background.jpg')
im.thumbnail((128,128))

im.show()
```

运行结果如图




生成对应图片的缩略图

3. 复制和粘贴图像区域

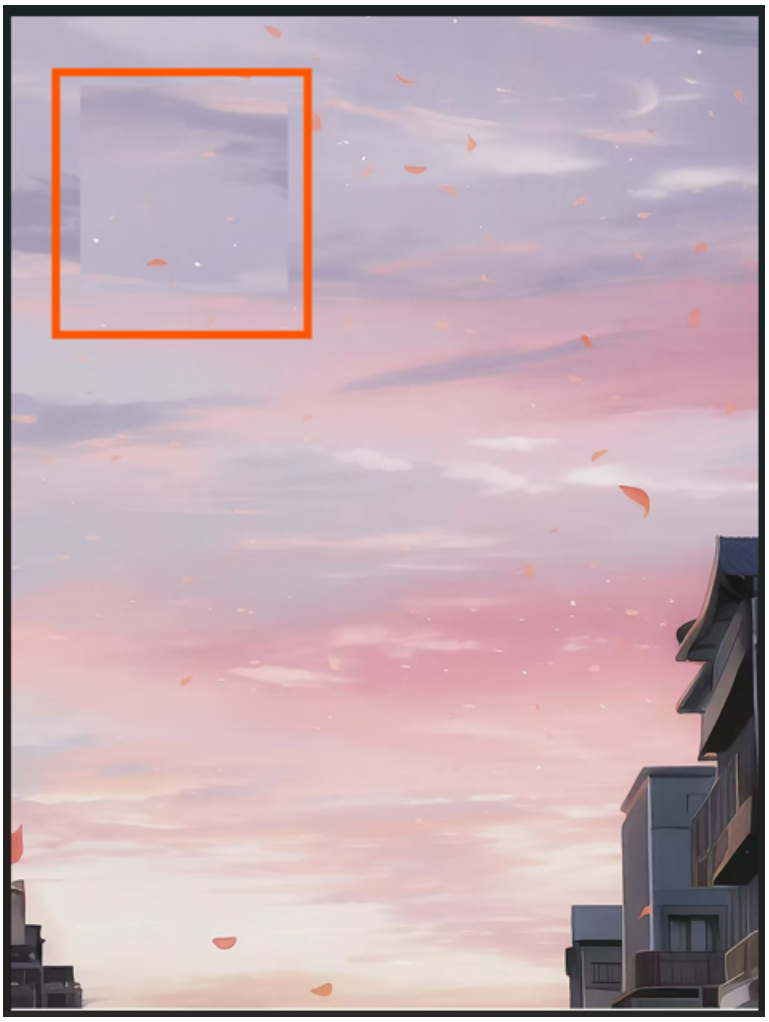
```
from PIL import Image

im = Image.open('background.jpg')

box = (100,100,400,400)
region = im.crop(box)
region = region.transpose(Image.ROTATE_180)
im.paste(region,box)

im.show()
```

运行结果如图



可以看到，图片中圈中部分被倒转后展示

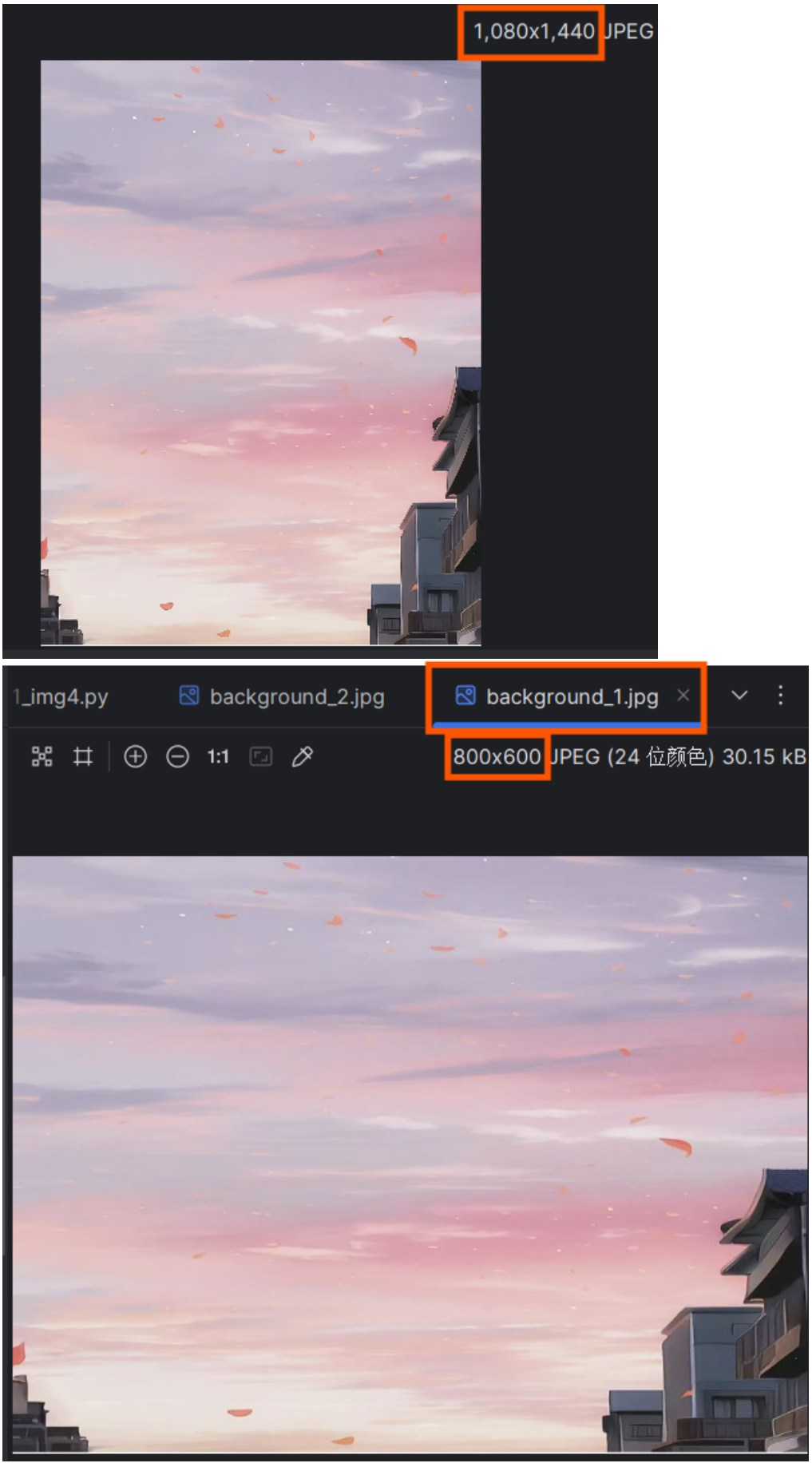
4. 调整一幅图像的尺寸

```
from PIL import Image
def resize_image(image_path, output_path, size):
    image = Image.open('background.jpg')
    resized_image = image.resize(size)
    resized_image.save('background_1.jpg')

resize_image('background.jpg', 'background_1.jpg', (800, 600))
```

运行结果如图

对比图片前后尺寸可以看到，尺寸被更改为对应尺寸 800 * 600

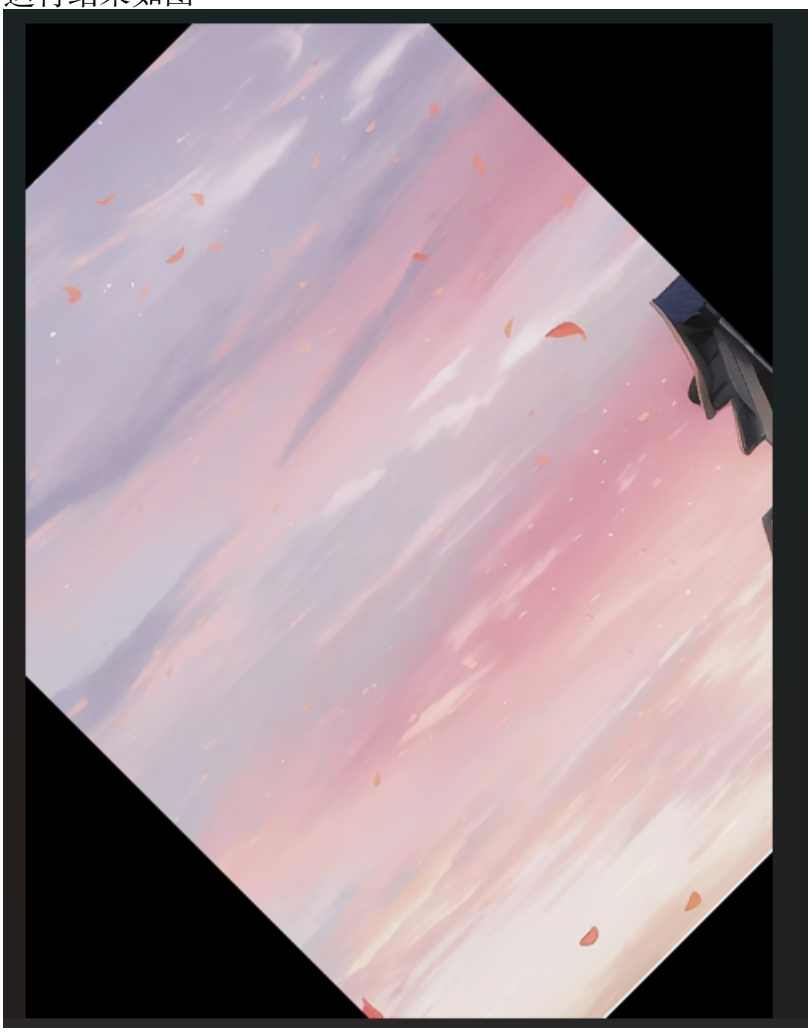

1,080x1,440 JPEG
1_img4.py
background_2.jpg
background_1.jpg
1:1
800x600 JPEG (24 位颜色) 30.15 kB

5. 旋转图像

```
from PIL import Image

im = Image.open('background.jpg')
out = im.rotate(45)
out.show()
```

运行结果如图



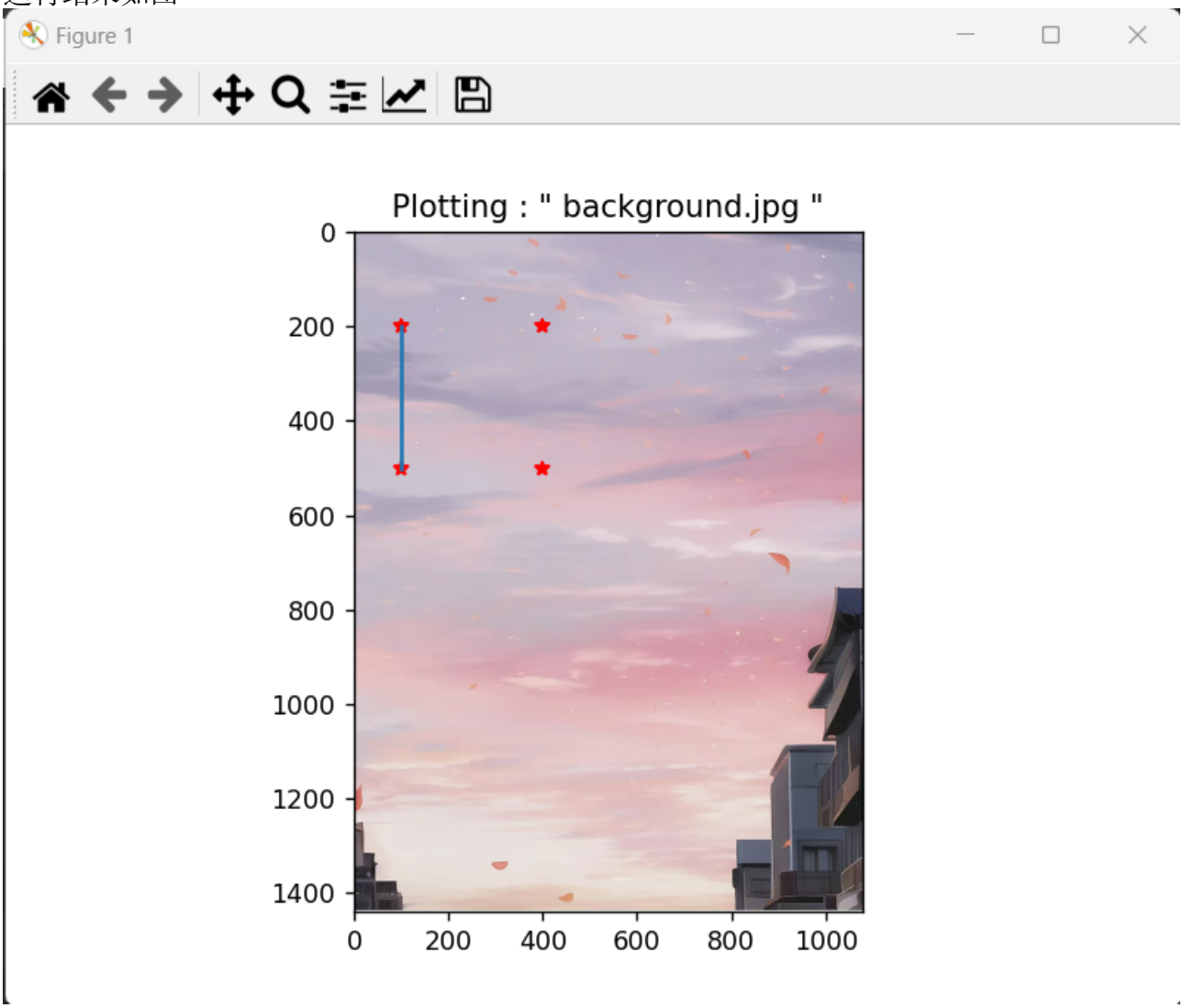
可以看到，图片被旋转 45° 展示

6. 绘制图像、点和线

```
from PIL import Image
from pylab import *
# 读 取 图 像 到 数 组 中
im = array(Image.open('background.jpg'))
```

```
# 绘制图像
imshow(im)
# 一些点
x = [100,100,400,400]
y = [200,500,200,500]
# 使用红色星状标记绘制点
plot(x, y, 'r*')
# 绘制连接前两个点的线
plot(x[:2], y[:2])
# 添加标题，显示绘制的图像
title ( ' Plotting : " background.jpg " ')
show ()
```

运行结果如图



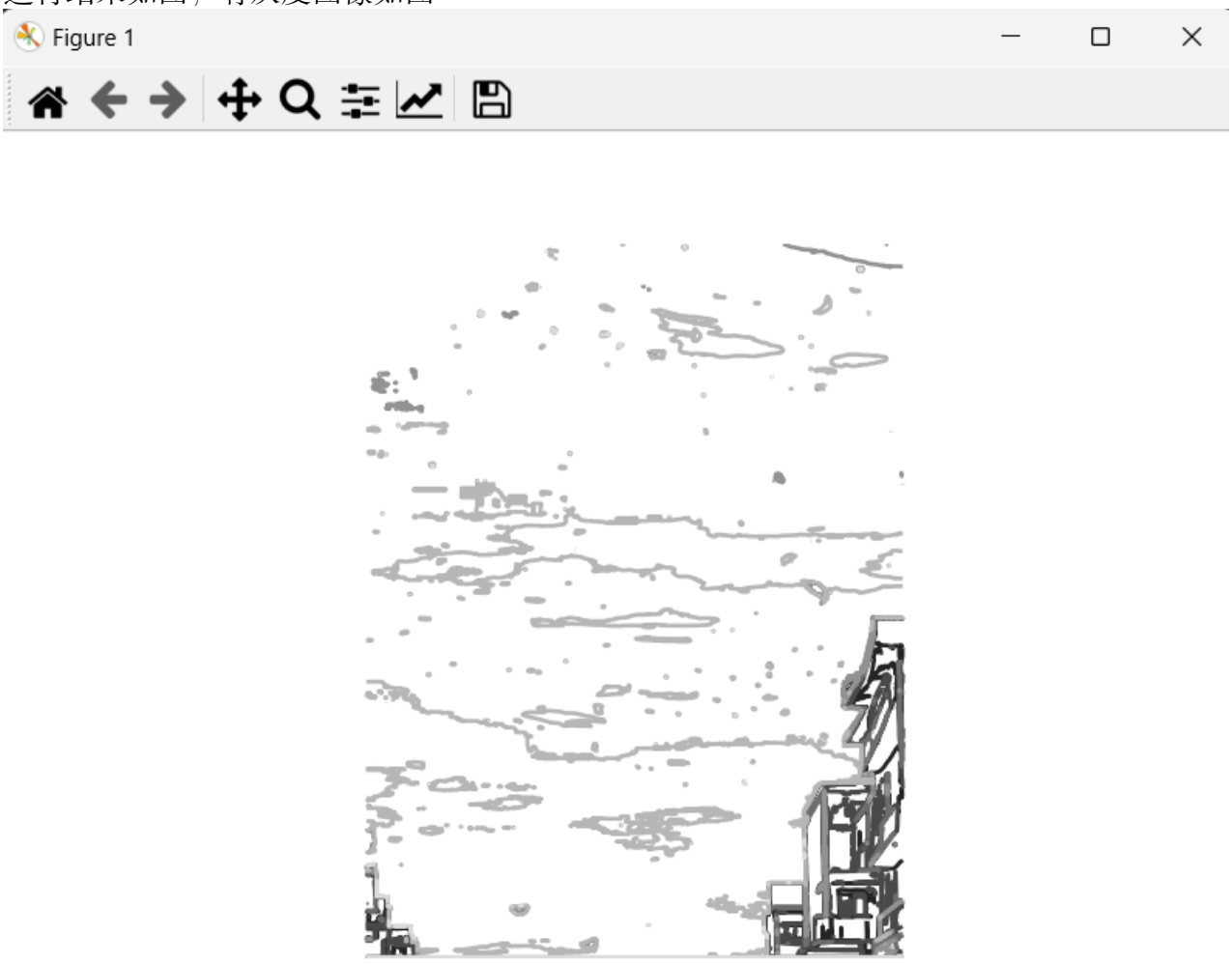

7. 绘制灰度图像

```
from PIL import Image
from pylab import *

im = array(Image.open('background.jpg').convert('L'))

gray()
contour(im, origin='image')
axis('equal')
axis('off')
show()
```

运行结果如图，有灰度图像如图




8. 绘制图像的直方图

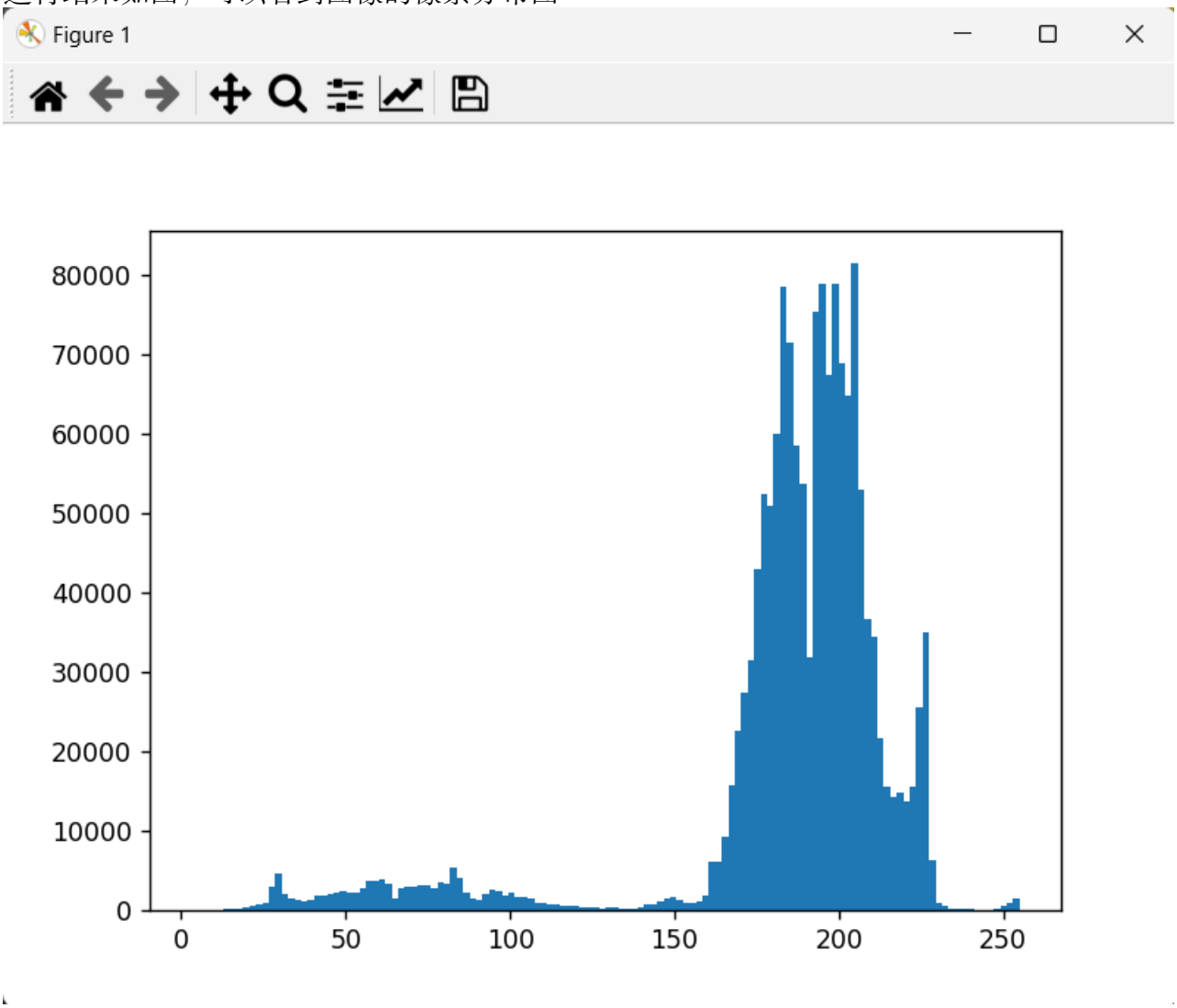
```
from PIL import Image
```

```
from pylab import *

im = array(Image.open('background.jpg').convert('L'))
figure()

hist(im.flatten(), 128)
show()
```

运行结果如图，可以看到图像的像素分布图




9. 将灰度图像进行反相处理，并查看图像中的最小和最大像素值

```
from PIL import Image
from numpy import *

im = array(Image.open('background.jpg').convert('L'))
im2 = 255 - im
```

```
print(int(im2.min()), int(im2.max()))
```

运行结果如图

```
D:\anaconda\python.exe D:\pycharmProjects\pythonProject\1_img2.py
0 252

进程已结束，退出代码为 0
```

10. 将图像的像素值变换到 100...200 区间，并查看图像中的最小和最大像素值

```
from PIL import Image
from numpy import *

im = array(Image.open('background.jpg').convert('L'))
im3 = (100.0/255) * im + 100
print(int(im3.min()), int(im3.max()))
```

运行结果如图

```
D:\anaconda\python.exe D:\pycharmProjects\pythonProject\1_img2.py
101 200

进程已结束，退出代码为 0
```

11. 对图像使用二次函数变换，使较暗的像素值变得更小，并查看图像中的最小和最大像素值

```
from PIL import Image
from numpy import *

im = array(Image.open('background.jpg').convert('L'))
im4 = 255.0 * (im/255.0)**2
print(int(im4.min()), int(im4.max()))
```

运行结果如图

```
D:\anaconda\python.exe D:\pycharmProjects\pythonProject\1_img2.py
0 255

进程已结束，退出代码为 0
```

12. 将一个包含像素数据的 NumPy 数组转换为 PIL.Image 对象

```
from PIL import Image
from numpy import *

im = array(Image.open('background.jpg').convert('L'))
im = Image.fromarray(uint8(im))

im.save('background_2.jpg')
```

运行结果如图

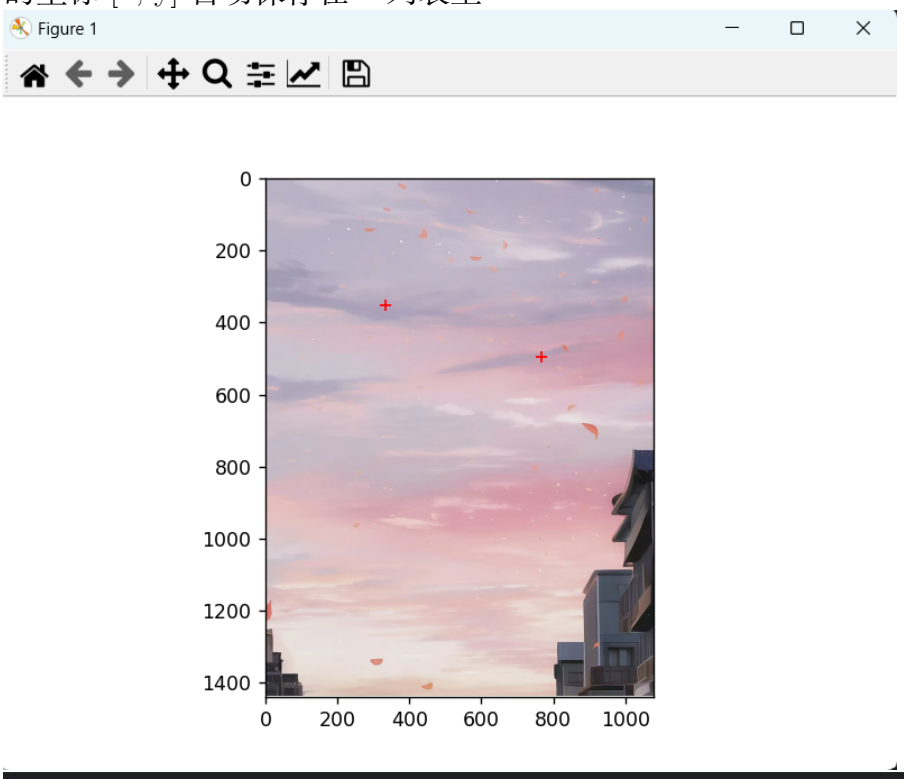
13. 使用 ginput() 函数实现交互式标注

```
from PIL import Image
from pylab import *

im = array(Image.open('background.jpg'))
imshow(im)
print('Please click 3 points')
x = ginput(3)
print('you clicked:', x)
show()
```

运行结果如图

绘制一幅图像，然后等待用户在绘图窗口的图像区域点击三次。程序将这些点击的坐标 [x, y] 自动保存在 x 列表里




```
D:\anaconda\python.exe D:\pycharmProjects\pythonProject\1_img1.py
Please click 3 points
you clicked: [(204.43506493506493, 265.9935064935064), (882.3571428571429, 421.8376623376623), (936.9025974025974, 1185.474025974026)]
```

# 3 实验总结与体会

通过学习命令行环境相关基础知识，了解了基本的进程控制、配置文件等操作的流程，熟悉了控制多进程等待或并行、远端设备控制等相关操作流程；通过学习 python 基础知识，掌握了 python 基本的逻辑语法与编写规则，并在此基础上学习了图像的识别与处理，初步认识了计算机视觉。

附课程仓库地址：`https://github.com/Physical2/systemToolsBase`（点击跳转）