

# 系统开发工具基础

## 命令行环境 & Python

王志巍 22020007161

2024 - 09 - 06

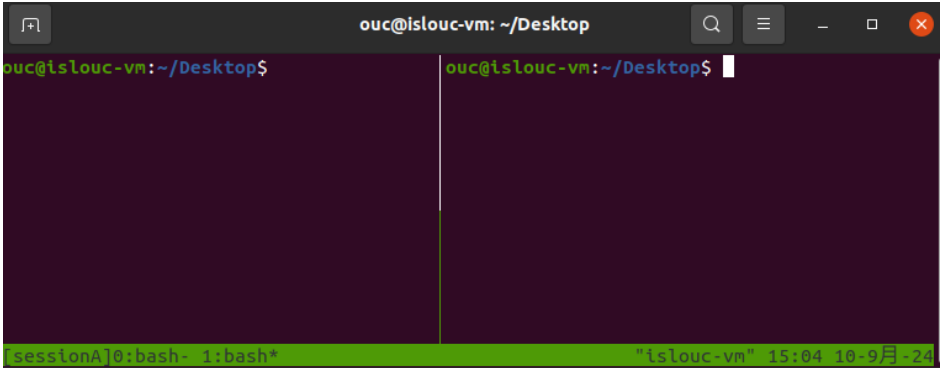
### 目录

|                           |           |
|---------------------------|-----------|
| <b>1 命令行环境</b>            | <b>1</b>  |
| 1.1 终端多路复用 tmux . . . . . | 1         |
| 1.2 别名 . . . . .          | 1         |
| 1.3 配置文件 . . . . .        | 1         |
| 1.4 远端设备 . . . . .        | 2         |
| 1.4.1 连接虚拟机 . . . . .     | 2         |
| 1.4.2 远程命令 . . . . .      | 3         |
| 1.4.3 文件交互 . . . . .      | 3         |
| <b>2 Python 入门基础</b>      | <b>4</b>  |
| 2.1 运算符 . . . . .         | 4         |
| 2.1.1 算术运算符 . . . . .     | 4         |
| 2.1.2 比较运算符 . . . . .     | 4         |
| 2.1.3 逻辑运算符 . . . . .     | 4         |
| 2.2 注释 . . . . .          | 4         |
| 2.3 输入输出 . . . . .        | 4         |
| 2.4 条件语句 . . . . .        | 5         |
| 2.5 循环语句 . . . . .        | 5         |
| 2.5.1 for 循环 . . . . .    | 5         |
| 2.5.2 while 循环 . . . . .  | 5         |
| 2.6 字典 . . . . .          | 5         |
| <b>3 Python 视觉应用</b>      | <b>7</b>  |
| 3.1 PIL . . . . .         | 7         |
| 3.2 Matplotlib . . . . .  | 8         |
| 3.3 NumPy . . . . .       | 9         |
| 3.4 SciPy . . . . .       | 10        |
| <b>4 实验心得</b>             | <b>11</b> |
| <b>5 Github 链接</b>        | <b>11</b> |

# 1 命令行环境

## 1.1 终端多路复用 tmux

```
tmux new -s session_name      # 创建新会话
ctrl+ b d                    # 分离会话
tmux ls                       # 列出所有会话
tmux attach -t session_name   # 重新连接到某个会话
ctrl+ b c                    # 创建新窗口
ctrl+ b %                    # 水平分割
ctrl+ b "                    # 垂直分割
```



## 1.2 别名

### 获取最常用的十条命令

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

排除个别 ssh 和程序调试指令，最常用的指令为：

```
python3
gcc -o
ls
ls -l
cd
```

### 设置别名

```
alias py='python3 '
alias gco='gcc -o '
alias sl='ls '
alias ll='ls -l '
alias dc='cd '
```

## 1.3 配置文件

将刚刚设置的别名添加到配置文件，保证其永久有效。

```
nano ~/.bashrc

alias py='python3 '
alias gco='gcc -o '
alias sl='ls '
alias ll='ls -l '
alias dc='cd '
```

```
source ~/.bashrc
```

```
ouc@islouc-vm:~/Desktop/terminal$ gcc testc test.c
ouc@islouc-vm:~/Desktop/terminal$ ./testc
Hello, World!
```

```
ouc@islouc-vm:~/Desktop/terminal$ python testpy.py
Hello, World!
```

## 1.4 远端设备

### 1.4.1 连接虚拟机

```
ssh -p 22 ouc@192.168.17.130
```

```
The authenticity of host '192.168.17.130 (192.168.17.130)' can't be established.
ECDSA key fingerprint is SHA256:88iH6zJ5rc14dTWtkEpzbjibmqKjztgsHdgPQFimFTA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.17.130' (ECDSA) to the list of known hosts.
ouc@192.168.17.130's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-107-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

51 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

11 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Your Hardware Enablement Stack (HWE) is supported until April 2025.
*** System restart required ***
Last login: Tue Sep 26 19:38:06 2023 from 192.168.155.91
ouc@islouc-vm:~$
```

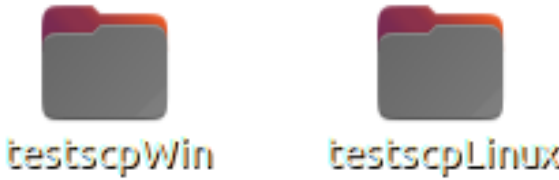
1.4.2 远程命令

```
ouc@islouc-vm:~$ ls
BurpSuiteCommunity Documents ghidra_10.3.3_PUBLIC Music Public Templates
Desktop Downloads idafree-8.3 Pictures snap Videos
ouc@islouc-vm:~$ cd Desktop/
ouc@islouc-vm:~/Desktop$ ls
exp5 'IDA Freeware 8.3.desktop' 'shell&vim' test
vmware-tools-distrib
exp7 ITSystem sql-labs-mysql.sh user
Ghidra.desktop Misc system VMwareTools-10.3.23-16594550.tar.gz
ouc@islouc-vm:~/Desktop$ cd shell&vim/
ouc@islouc-vm:~/Desktop/shell&vim$ ll
total 84
drwxrwxr-x 6 ouc ouc 4096 9月 3 19:01 ./
drwxr-xr-x 10 ouc ouc 4096 9月 3 15:00 ../
-rw-rw-r-- 1 ouc ouc 2028 9月 3 17:41 all
-rw-rw-r-- 1 ouc ouc 205 9月 3 11:23 execl.sh
-rw-rw-r-- 1 ouc ouc 195 9月 3 11:23 exec.gz
-rw-rw-r-- 1 ouc ouc 128 9月 3 11:06 exec.sh
-rw-rw-r-- 1 ouc ouc 78 9月 3 12:05 fetch.sh
-rw-rw-r-- 1 ouc ouc 186 9月 3 11:45 html2zip.sh
-rw-rw-r-- 1 ouc ouc 1006 9月 3 11:46 htmlFiles.zip
-rwxrwxrwx 1 ouc ouc 168 8月 30 09:23 ll.sh*
-rwxrwxrwx 1 ouc ouc 346 8月 30 09:30 marco.sh*
-rw-rw-r-- 1 ouc ouc 450 9月 3 10:41 random.sh
-rw-rw-r-- 1 ouc ouc 429 9月 3 10:42 script_output.log
-rw-rw-r-- 1 ouc ouc 21 9月 3 19:01 sebSwap
-rw-rw-r-- 1 ouc ouc 24 9月 3 18:58 sebSwap.bak
drwxrwxr-x 2 ouc ouc 4096 9月 3 09:58 test/
drwxrwxr-x 2 ouc ouc 4096 9月 3 11:06 testExec/
drwxrwxr-x 3 ouc ouc 4096 9月 3 11:36 testXargs/
-rw-rw-r-- 1 ouc ouc 1695 9月 3 17:46 unused
-rw-rw-r-- 1 ouc ouc 333 9月 3 17:42 used
drwxrwxr-x 3 ouc ouc 4096 9月 3 16:55 vims/
```

1.4.3 文件交互

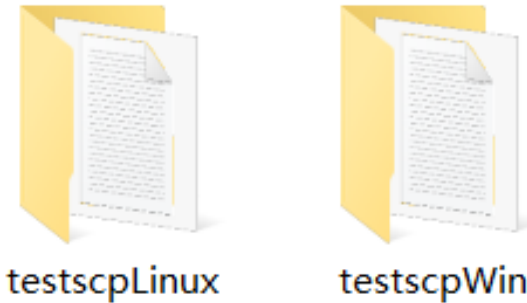
上传

```
scp -r C:\Users\Starry_sky\Desktop\开发工具\gitTest\0906\testscpWin
ouc@192.168.17.130:~/Desktop
```



下载

```
# 指令末尾的.是-r的第二个参数
# 表示将远程服务器上的目录复制到本地的当前目录
scp -r ouc@192.168.17.130:~/Desktop/testscpLinux .
```



## 2 Python 入门基础

### 2.1 运算符

#### 2.1.1 算术运算符

```
print(a + b) # 加法
print(a - b) # 减法
print(a * b) # 乘法
print(a / b) # 除法
print(a % b) # 取模
print(a ** b) # 幂运算
print(a // b) # 整除
```

#### 2.1.2 比较运算符

```
print(a == b) # 等于
print(a != b) # 不等于
print(a > b) # 大于
print(a < b) # 小于
print(a >= b) # 大于等于
print(a <= b) # 小于等于
```

#### 2.1.3 逻辑运算符

```
print(a and b) # 与
print(a or b) # 或
print(not a) # 非
```

### 2.2 注释

```
# 单行注释
print("hello world")
'''
多行注释
'''
print("hello world")
```

### 2.3 输入输出

input() 函数用于从用户那里获取输入。它会暂停程序的执行，等待用户输入，并在用户按下回车键后继续执行。输入的数据会作为字符串返回。函数可以接受至多一个参数，作为提示信息输出。

split() 方法是对于字符串的处理，将字符串按照指定的分隔符拆分成一个列表。由于 input() 将一整行作为字符串读入，因此常常配合 split() 进行输入处理。

```
numbers = input().split(',')
for num in numbers:
    print(num)
```

输入

```
1,2,3,4,5
```

输出

```
1
2
3
4
5
```

## 2.4 条件语句

```
number = int(input("输入一个数字："))
if number > 0:
    print("正数")
elif number < 0:
    print("负数")
else:
    print("零")
```

## 2.5 循环语句

### 2.5.1 for 循环

for 循环可以通过”for i in range(a, b):” 循环一定次数。

```
num = int(input("输入一个正整数："))
if num > 1:
    for i in range(2, int(num ** 0.5) + 1):
        if (num % i) == 0:
            print(num, "不是素数")
            break
    else:
        print(num, "是素数")
else:
    print(num, "不是素数")
```

也可以通过”for num in numbers:” 对列表进行循环。

```
numbers = input().split(',')
for num in numbers:
    print(num)
```

### 2.5.2 while 循环

```
i = 1
while i <= 10:
    print(i)
    i += 1
```

## 2.6 字典

在 Python 中，字典 (dictionary) 是一种可变的、无序的键值对集合。每个键 (key) 与一个值 (value) 相关联，键必须是唯一的且不可变的（如字符串、数字或元组），而值可以是任何数据类型。

可以使用花括号 或 dict() 函数来创建字典。

```
dict = {
    "name": "wzw",
    "school": "OUC"
```

```
}  
# 或  
dict = dict(name="wzw", scholl="OUC")
```

可以使用 for 循环遍历字典的键、值或键值对。

```
# 键  
for key in dict:  
    print(key)  
# 值  
for value in dict.values():  
    print(value)  
# 键值对  
for key, value in dict.items():  
    print(f"{key}: {value}")
```

字典提供了许多有用的方法，如 get(), keys(), values(), items(), pop(), update()

```
# 使用键获取值  
print(dict.get("name"))  
# 获取所有键  
print(dict.keys())  
# 获取所有值  
print(dict.values())  
# 获取所有键值对  
print(dict.items())  
# 使用 pop() 方法删除元素  
name = dict.pop("name")  
# 使用 update() 方法更新字典  
dict.update({"name": "Bob", "school": "unknown"})
```

**对一个字典进行增，删，改，查**

```
students = {}  
def add_student(student_id, name):  
    if student_id in students:  
        print(f"{student_id} exist")  
    else:  
        students[student_id] = name  
        print("success")  
def delete_student(student_id):  
    if student_id in students:  
        del students[student_id]  
        print("success")  
    else:  
        print(f"{student_id} not found")  
def update_student(student_id, name):  
    if student_id in students:  
        students[student_id] = name  
        print("success")  
    else:  
        print(f"{student_id} not found")  
def get_student(student_id):  
    if student_id in students:
```

```

        print(f"{student_id}: {students[student_id]}")
    else:
        print(f"{student_id} not found")
def display_all_students():
    if students:
        print("All students:")
        for student_id, name in students.items():
            print(f"{student_id}: {name}")
    else:
        print("None")
add_student("202401", "Tom")
add_student("202402", "Jerry")
display_all_students()
update_student("202401", "Bob")
get_student("202401")
delete_student("202402")
display_all_students()

```

## 输出

```

success
success
All students:
202401: Tom
202402: Jerry
success
202401: Bob
success
All students:
202401: Bob

```

## 3 Python 视觉应用

### 3.1 PIL

PIL (Python Imaging Library) 是一个用于图像处理的库。它提供的功能包括打开、操作和保存各种格式的图像文件。目前 PIL 库已经被 Pillow 库所取代。

```

from PIL import Image
# 打开图片
image = Image.open("image.jpg")
# 处理前
image.show()
# 缩放
new_size = (800, 600) # 新的尺寸
resized_image = image.resize(new_size)
# 旋转
angle = 45 # 旋转角度
rotated_image = resized_image.rotate(angle)
# 转换为RGB模式
rgb_image = rotated_image.convert("RGB")
# 处理后
rgb_image.show()
# 保存处理后的图片

```



```
rgb_image.save("image2.jpg", format="JPEG")
```



原图



旋转 45°

最开始，我一直使用的是 JPEG 格式的 PIL 变量，但是在最后保存时遇到问题，“conversion from RGB to JPEG not supported”。查询资料发现 JPEG 虽然能查看 RGB，但是 JPEG 的 convert 方法不支持 RGB 到 JPEG 的转换。因此在 PIL 处理完图像后，先将其存为 RGB，再使用 RGB 的 convert 方法，将其转换为 JPEG。

### 3.2 Matplotlib

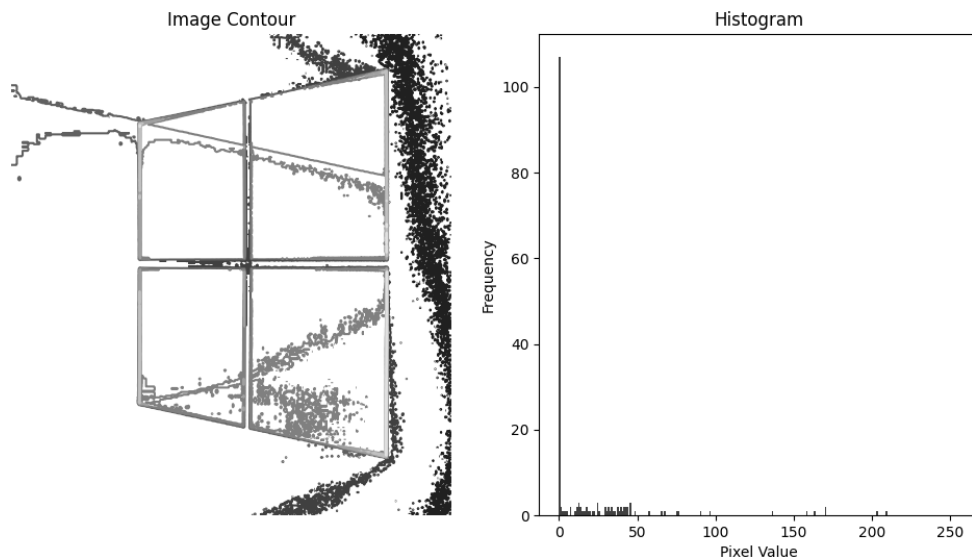
```
from PIL import Image
import matplotlib.pyplot as plt

image = Image.open("image.jpg")
# 转换为灰度图
gray_image = image.convert("L")
# 获取图像像素数据
pixels = list(gray_image.getdata())
width, height = gray_image.size
pixels = [pixels[i * width:(i + 1) * width] for i in range(height)]

# 图像轮廓
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.title("Image Contour")
plt.contour(pixels, cmap='gray')
plt.axis('off')

# 图像直方图
plt.subplot(1, 2, 2)
plt.title("Histogram")
plt.hist(gray_image.histogram(), bins=256,\
         range=[0, 256], color='black', alpha=0.75)
plt.xlabel('Pixel Value')
plt.ylabel('Frequency')

# 保存图像
plt.savefig("plt.png")
# 显示图像
plt.show()
```



### 3.3 NumPy

NumPy (Numerical Python) 提供了支持大规模多维数组和矩阵运算的功能，并且包含了大量的数学函数库，用于对这些数组进行操作。主要功能有：多维数组对象处理、不同形状的数组之间算术运算、向量化运算、线性代数函数库等。

由于图像本质上是由像素值组成的多维数组，而 NumPy 提供了高效的多维数组操作功能，使得图像处理变得更加便捷高效。

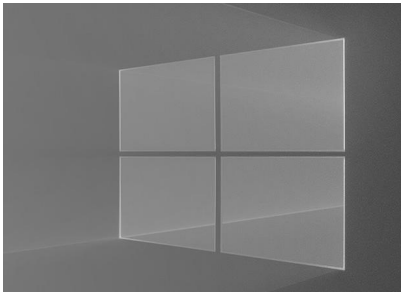
```
import numpy as np
from PIL import Image
import cv2

image = Image.open("image.jpg")
image_array = np.array(image)
# 灰度操作
gray_image = image.convert("L")
gray_image_array = np.array(gray_image)

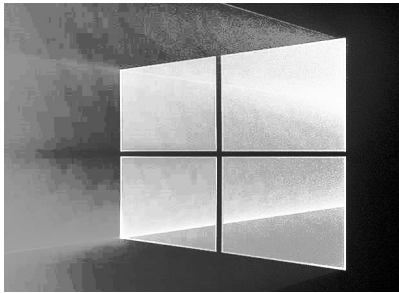
# 灰度图直方图均衡化
equalized_gray_image_array = \
    cv2.equalizeHist(gray_image_array)

# 原图直方图均衡化 (HSV)
image_hsv = cv2.cvtColor(image_array, cv2.COLOR_RGB2HSV)
image_hsv[:, :, 2] = cv2.equalizeHist(image_hsv[:, :, 2])
equalized_color_image_array = \
    cv2.cvtColor(image_hsv, cv2.COLOR_HSV2RGB)

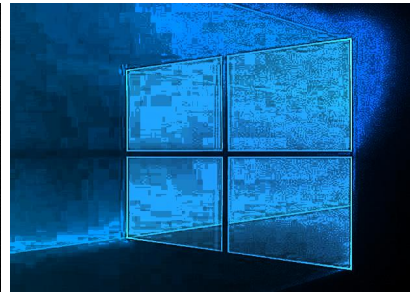
# 保存图像
gray_image_pil = Image.fromarray(gray_image_array)
gray_image_pil.save("gray_image.jpg")
equalized_gray_image_pil = \
    Image.fromarray(equalized_gray_image_array)
equalized_gray_image_pil.save("equalized_gray_image.jpg")
equalized_color_image_pil = \
    Image.fromarray(equalized_color_image_array)
equalized_color_image_pil.save("equalized_color_image.jpg")
```



灰度图



灰度图直方图均衡化



原图直方图均衡化

### 3.4 SciPy

```
import numpy as np
from PIL import Image
from scipy.ndimage import gaussian_filter, sobel

image = Image.open("image.jpg")
image_array = np.array(image)

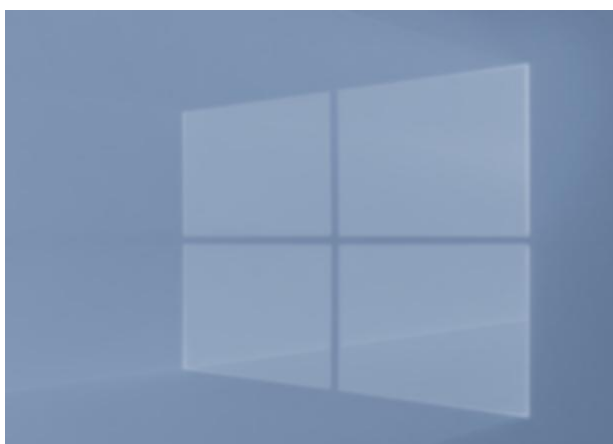
# 图像模糊（高斯滤波）
blurred_image_array = gaussian_filter(image_array, sigma=2)

if image.mode == 'RGBA':
    blurred_image_array = blurred_image_array[:, :, :3]

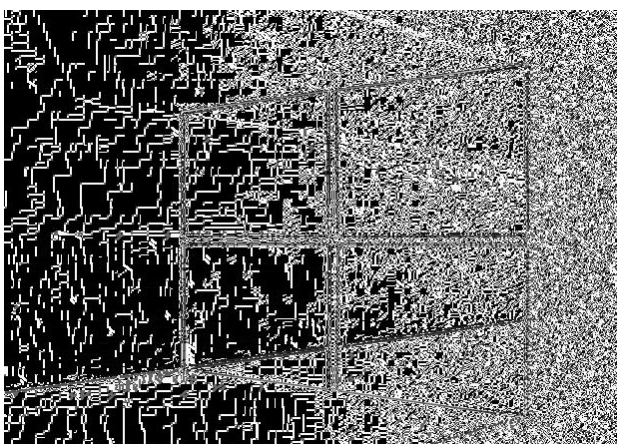
# 灰度转换
gray_image_array = \
    np.array(Image.open("image.jpg").convert("L"))

# 图像导数（Sobel算子）
sobel_x = sobel(gray_image_array, axis=0)
sobel_y = sobel(gray_image_array, axis=1)
sobel_image_array = np.hypot(sobel_x, sobel_y)

# 保存处理后的图像
blurred_image_pil = \
    Image.fromarray(blurred_image_array.astype(np.uint8))
blurred_image_pil.save("blurred_image.jpg")
sobel_image_pil = \
    Image.fromarray(sobel_image_array.astype(np.uint8))
sobel_image_pil.save("sobel_image.jpg")
```



图像模糊



图像导数

## 4 实验心得

通过命令行环境的实验，了解了更多远端交互的操作，运用到主机和虚拟机间，使得日常的交互方便了许多。此外，学习了 Python 基础和图像处理相关的 Python 库，对这方面的知识有了更多了解。

## 5 Github 链接

```
https://github.com/Starry-Sky-OUC/gitTest
```