



中国海洋大学
OCEAN UNIVERSITY OF CHINA

第三周

实 验 报 告

学 院：信息科学与工学部

班 号：计科一班

姓 名：顾晓宁

学 号：24020007036

实验编号：第三周实验报告

指导教师：周小伟

目录

1 实验目的	3
2 练习内容与结果	3
2.1 Python基础	3
2.1.1 list:创建、访问、修改	3
2.1.2 list:+ * list()	3
2.1.3 list:函数若干	3
2.1.4 list:方法若干	4
2.2 py图形库基础	4
2.2.1 Pillow库:打开、访问、展示、保存?	4
2.2.2 Pillow库: 转换图像模式	5
2.2.3 Pillow库:调整图像大小	5
2.2.4 Pillow库:裁剪图像	5
2.2.5 Pillow库:旋转和翻转	6
2.2.6 Pillow库:图像粘合	6
2.2.7 Pillow库:图像增强	6
2.3 命令行环境	7
2.3.1 如何停止作业/进程?	7
2.3.2 如何暂停进程, 暂停后可以进行那些操作?	7
2.3.3 tmux的会话的使用?	8
2.3.4 tmux的窗口使用?	8
2.3.5 tmux的面板使用?	9
2.3.6 别名alias的使用与配置?	9
2.3.7 ssh安装与使用?	11
2.3.8 ssh密钥?	11
2.3.9 scp复制文件?	12
3 实验感悟	12
4 个人github/gitee账号	12
5 友情链接	12

第三周实验：命令行环境、python基础、py图形库

1 实验目的

学习配置基本的命令行环境，学习如何使用ssh连接服务器，以及python基础及其图形库

2 练习内容与结果

2.1 Python基础

[查看文件](#)

因为具有一定的py基础，部分内容不附上，可参见参考资料页的连接。

[这是一部分练习用例，有关list](#)

2.1.1 list:创建、访问、修改

创建：

```
list()
```

```
var = [e[,e...]]
```

等

访问：

使用[]下标索引可访问元素，支持负数索引

可以使用切片[start:end:step]

修改略

2.1.2 list:+ * list()

+ 拼接

* 重复，注意浅拷贝等问题

list() 转化为list

2.1.3 list:函数若干

```
1 len(list)
```

列表元素个数

```
2 max(list)
```

返回列表元素最大值

```
3 min(list)
```

返回列表元素最小值

4 `list(seq)`

Python包含以下方法:

2.1.4 list:方法若干

1 `list.append(obj)`

在列表末尾添加新的对象

2 `list.count(obj)`

统计某个元素在列表中出现的次数

3 `list.extend(seq)`

在列表末尾一次性追加另一个序列中的多个值（用新列表扩展原来的列表）

4 `list.index(obj)`

从列表中找出某个值第一个匹配项的索引位置

5 `list.insert(index, obj)`

将对象插入列表

6 `list.pop([index=-1])`

移除列表中的一个元素（默认最后一个元素），并且返回该元素的值

7 `list.remove(obj)`

移除列表中某个值的第一个匹配项

8 `list.reverse()`

反向列表中元素

9 `list.sort(cmp=None, key=None, reverse=False)`

对原列表进行排序

2.2 py图形库基础

[基础语法](#)

以下涉及的Pillow库的基本演示，可以见该目录下的py-code-test文件夹，内附有demo。

[代码文件](#)

2.2.1 Pillow库:打开、访问、展示、保存?

1. 打开和创建图像

打开现有图像，参数是路径:

使用 `Image.open()` 函数。

创建新图像:

使用 `Image.new()` 函数。

例如

```
new_img = Image.new('RGB', (400, 300), color='red')
```

2. 图像的基本属性

即类Image的属性

```
format      # 图像格式 (e.g., JPEG, PNG)
size        # 图像尺寸 (宽度, 高度)
width       # 图像宽度
height      # 图像高度
mode        # 颜色模式 (e.g., RGB, RGBA, L (灰度), CMYK)
```

3. 显示图像

`img.show()` # 会使用系统默认的图片查看器打开图像

4. 保存图像

使用 `save()` 方法, 可以指定格式和质量等参数。

```
img.save('new_image.png')      # 根据扩展名自动判断格式
img.save('output.jpg', 'JPEG') # 明确指定格式
img.save('high_quality.jpg', quality=95) # 保存为高质量JPEG (1-100, 默认75)
img.save('small_size.png', optimize=True) # 优化PNG文件大小
```

2.2.2 Pillow库: 转换图像模式

使用方法`convert`

```
gray_img = img.convert('L')    # 转换为灰度图像
rgba_img = img.convert('RGBA') # 转换为带透明通道的图像
```

2.2.3 Pillow库:调整图像大小

1.`resize()` 方法:

缩放到指定尺寸

```
resized_img = img.resize((400, 300))
```

高质量缩放推荐使用 `Lanczos` 滤波器

```
resized_img = img.resize((400, 300), Image.LANCZOS)
```

2.`thumbnail()` 方法:

原地修改, 生成保持原图宽高比的缩略图

尺寸参数是一个元组, 表示最大宽度和高度。

例如:

```
img.thumbnail((150, 150)) # 图像的宽或高将被限制在150像素以内, 比例不变
```

2.2.4 Pillow库:裁剪图像

使用 `crop()` 方法, 参数是一个定义左、上、右、下像素坐标的元组 (`left`, `upper`, `right`, `lower`)。

注意右下坐标分别大于左上，原点为图像左上角。

2.2.5 Pillow库:旋转和翻转

1. 旋转 rotate()

旋转45度，并扩展画布以避免图像被裁剪，背景填充为白色

```
rotated_img = img.rotate(45, expand=True, fillcolor='white')
```

2. 翻转 transpose()

水平翻转

```
flipped_img = img.transpose(Image.FLIP_LEFT_RIGHT)
```

垂直翻转

```
flipped_img = img.transpose(Image.FLIP_TOP_BOTTOM)
```

2.2.6 Pillow库:图像粘合

使用 `paste()` 方法可以将一张图像粘贴到另一张图像上。

`paste()` 会原地修改 `img1`。

假设有 `img1` 和 `img2`

将 `img2` 粘贴到 `img1` 的 `(x, y)` 坐标处

```
img1.paste(img2, (50, 50))
```

可以作为水印

2.2.7 Pillow库:图像增强

需要先导入对应模块，

```
from PIL import ImageEnhance
```

增强对比度

```
enhancer = ImageEnhance.Contrast(img)
```

```
enhanced_img = enhancer.enhance(2.0) # 增强2倍
```

增强亮度

```
enhancer = ImageEnhance.Brightness(img)
```

```
bright_img = enhancer.enhance(1.5) # 增强1.5倍
```

增强色彩饱和度

```
enhancer = ImageEnhance.Color(img)
```

```
colorful_img = enhancer.enhance(1.5)
```

增强锐度

```
enhancer = ImageEnhance.Sharpness(img)
```

```
sharp_img = enhancer.enhance(2.0)
```

2.3 命令行环境

[文字版](#)

2.3.1 如何停止作业/进程?

一般可以使用`ctrl+c` 或者 `ctrl+\` 或者 `kill -TERM PID`, 来结束一个进程, 也可使用`kill` 作业号, 来结束一个作业。

核心转储是指接受`SIGQUIT`信号/程序意外终止, 将其快照存为一个文件, 通常命名为`core.PID`, 可用以调试。或调用`apport`自动处理崩溃报告。

```
sufwis@vmubuntu:~/learn$ sleep 1000&
[1] 3061
sufwis@vmubuntu:~/learn$ jobs -l
[1]+  3061 运行中                  sleep 1000 &
sufwis@vmubuntu:~/learn$ kill -TERM 3061
sufwis@vmubuntu:~/learn$ jobs
[1]+  已终止                      sleep 1000
sufwis@vmubuntu:~/learn$
```

图 1: `kill -TERM` 如图

```
sufwis@vmubuntu:~/learn$ sleep 1000
^C
sufwis@vmubuntu:~/learn$ sleep 1000
^退出 (核心已转储)
sufwis@vmubuntu:~/learn$ sleep 1000 &
[1] 3129
sufwis@vmubuntu:~/learn$ kill %1
sufwis@vmubuntu:~/learn$ jobs
[1]+  已终止                      sleep 1000
```

图 2: 其他结束方式如图

2.3.2 如何暂停进程, 暂停后可以进行那些操作?

`ctrl+z`可暂停并将之转为后台, 使用`jobs`查看后台的作业情况, `%+作业号`选取作业。之后使用`fg`或`bg`将之转为前台进行或后台进行。

`kill -STOP 作业号`, 暂停; `kill -CONT 作业号`, 继续进行;

使用命令+ &直接在后台运行，不过此时仍旧为shell子进程，shell推出后，都会结束运行。在输入命令时使用nohup，或者使用disown分离一个作业，这样在该shell结束时，该作业不会结束。

```
sufwis@vmubuntu:~$ (sleep 20;ls)&
[1] 3954
sufwis@vmubuntu:~$ jobs
[1]+  运行中                  ( sleep 20; ls --color=auto ) &
sufwis@vmubuntu:~$ disown %1
sufwis@vmubuntu:~$ jobs
sufwis@vmubuntu:~$ 公共  视频  文档  音乐  certifications  learn  softwares
模板  图片  下载  桌面  cmd_line_test.sh  snap
```

图 3: disown将作业从当前shell中移除如图

2.3.3 tmux的会话的使用?

tmux或tmux new -s name创建一个会话，tmux ls 查看会话，tmux a -t name 连接会话，倘若已连接，jC-b+d分离会话，返回终端，tmux -kill-session -t name 删除会话

```
sufwis@vmubuntu:~/learn$ tmux ls
0: 1 windows (created Sun Sep 14 09:34:43 2025)
1: 1 windows (created Sun Sep 14 09:40:49 2025)
third session: 1 windows (created Sun Sep 14 09:54:22 2025)
sufwis@vmubuntu:~/learn$ tmux kill-session -t 0
sufwis@vmubuntu:~/learn$ tmux ls
1: 1 windows (created Sun Sep 14 09:40:49 2025)
third session: 1 windows (created Sun Sep 14 09:54:22 2025)
```

图 4: 删除会话如图

2.3.4 tmux的窗口使用?

- <C-b> c 创建一个新的窗口，使用 <C-d> 关闭
- <C-b> N 跳转到第 N 个窗口，注意每个窗口都是有编号的
- <C-b> p 切换到前一个窗口
- <C-b> n 切换到下一个窗口
- <C-b> , 重命名当前窗口
- <C-b> w 列出当前所有窗口


```
(0) - 1: 4 windows (attached)
(1)  └─> 0: defaule*
(2)  └─> 1: bash
(3)  └─> 2: bash
(4)  └─> 3: bash-
(5) - third session: 1 windows
(6)  └─> 0: bash*
```

图 5: 列出当前所有窗口如图

2.3.5 tmux的面板使用?

面板是 `tmux` 中最小的工作区域，每个面板都可以独立运行一个进程，一个伪终端

`<C-b> "` 水平分割

`<C-b> %` 垂直分割

`<C-b> <方向>` 切换到指定方向的面板，`<方向>` 指的是键盘上的方向键

`<C-b> z` 切换当前面板的缩放

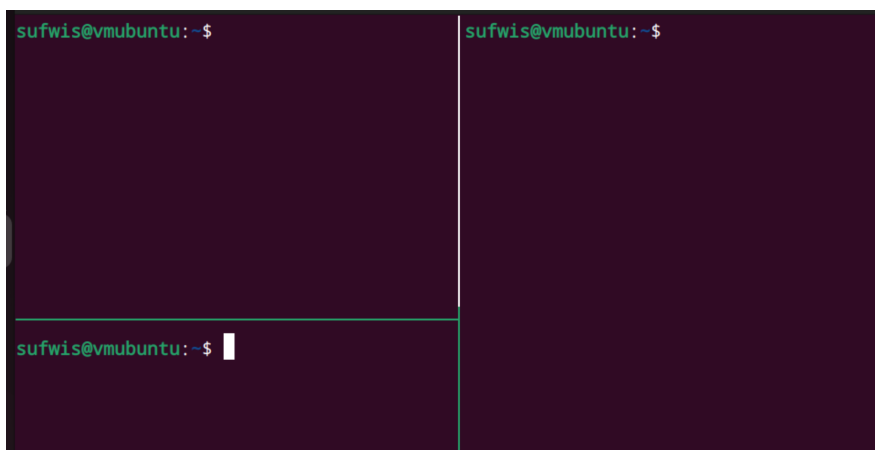


图 6: 使用垂直分割和水平分割如图

2.3.6 别名alias的使用与配置?

```
alias alias_name="command_to_alias arg1 arg2"
```

即可设定临时别名，如果放入`.bashrc`或对应的配置文件中，可以永久使用。

```
sufwis@vmubuntu:~/learn$ alias t=tree
sufwis@vmubuntu:~/learn$ t
.
├── bashlearn
│   ├── bashDebug
│   │   ├── eq32.sh
│   │   ├── serr.log
│   │   ├── sout.log
│   │   └── test.sh
│   └── bashfiles
│       ├── basictest.bash
│       ├── funtest.bash
│       ├── marco.sh
│       ├── other.bash
│       ├── test.sh
│       └── var.sh
```

图 7: 临时别名如图

假设使用tmux开启另一个shell进程，显示无该命令。

```
sufwis@vmubuntu:~/learn$ t
t: 未找到命令
sufwis@vmubuntu:~/learn$
```

图 8: 如图

在.bashrc中加入别名配置，然后在另一个终端中使用。

```
# my aliases
alias t=tree
```

图 9: 配置如图

```

sufwis@vmubuntu:~$ source ~/.bashrc
sufwis@vmubuntu:~$ t
.
├── 公共
├── 模板
├── 视频
├── 图片
├── 文档
├── 下载
├── 音乐
└── 桌面

```

图 10: 使用如图

2.3.7 ssh安装与使用?

```

1  # linux server
2  sudo apt install openssh-server # load
3  sudo systemctl start ssh # start ssh-server
4  sudo systemctl enable ssh # Set as self starting upon startup
5  sudo systemctl status ssh # check whether is started
6
7  # client
8  ssh username@host # then enter pwd

```

2.3.8 ssh密钥?

```

1  ssh-keygen
2  ssh-copy-id -i ~/.ssh/id_rsa.pub ubuntu@192.168.1.105

```

之后默认生成在~/.ssh/id_rsa, 或用户名/.ssh/id_rsa (Windows)。

id_rsa为私钥, id_rsa.pub: 公钥 (可公开, 用于配置到远程服务器),

将公钥拷贝到你要连接的服务器的.ssh/authorized_keys 内, 即可无需密码登录。

如果未使用默认路径, 可在对应命令处使用-i指明密钥的路径。

如果无法使用ssh-copy-id可以手动复制。

```

PS C:\Users\suffe> ssh sufwis@vmubuntu
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-29-generic x86_64)

```

图 11: 密钥无密码登录如图

2.3.9 scp复制文件？

`scp [本地文件路径] [远程用户]@[远程IP或域名]:[远程目标路径]`

`scp [远程用户]@[远程IP或域名]:[远程文件路径] [本地目标路径]`

SCP命令总是在源文件所在的那一端执行

SCP命令的执行位置决定了它能访问的文件系统，SCP命令总是在包含源文件的那一端执行

```
PS C:\Users\suffe> scp D:\temp_to_linux\name.txt sufwis@vmubuntu:/home/sufwis/learn/temp
name.txt
100% 0 0.0KB/s 00:00
PS C:\Users\suffe> |
```

图 12: win文件转Linux虚拟机如图

```
PS C:\Users\suffe> scp sufwis@vmubuntu:/home/sufwis/learn/temp/tmp.txt D:\temp_to_linux
PS C:\Users\suffe> dir D:\temp_to_linux

目录: D:\temp_to_linux

Mode                LastWriteTime         Length Name
----                -
-a----          2025/9/14   10:57             0 name.txt
-a----          2025/9/14   11:26             0 tmp.txt
```

图 13: Linux虚拟机文件转win 如图

3 实验感悟

python越来越流行，学习人数也越来越多。相对而言，python的语法简单，虽然运行效率相对低，但是相应的开发效率却很高。这离不开python多样的库、模块等。

ssh的使用同样普遍，例如连接云服务器。总之学无止境。

4 个人github/gitee账号

[对应仓库](#)

[个人github账号](#)

5 友情链接

[命令行环境](#)

[Python 基础教程](#)

[【Python】推荐五个常用的图像处理库](#)