

# 实验报告 3

姓名：李燮珍

学号：22020007048

实验时间：2024/09/06

## 一. 实验板块

命令行环境

python 入门基础

python 视觉应用

## 二. 实验内容

### 1. 命令行环境相关：

1. 任务控制
2. 结束进程
3. 暂停和后台执行进程
4. 终端多路复用
5. 别名
6. 配置文件 (Dotfiles)
7. 可移植性
8. 远端设备
9. Shell & 框架
10. 终端模拟器

## 2. 学习 python 入门基础知识

## 3. 学习 python 视觉应用相关

# 三. 实验过程

命令行环境相关实验：

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

```
lxz@lxz-virtual-machine:~/文档/system-development-tool$ mkdir ~/glts/dotfiles
mkdir: 无法创建目录“/home/lxz/glts/dotfiles”: 没有那个文件或目录
lxz@lxz-virtual-machine:~/文档/system-development-tool$ sleep 10000
^Z
[1]+  已停止                  sleep 10000
lxz@lxz-virtual-machine:~/文档/system-development-tool$ bg
^Z
LibreOffice Calc  -machine:~/文档/system-development-tool$
```

图 1: Caption

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`

```
lxz@lxz-virtual-machine:~/文档/system-development-tool$ mkdir ~/glts/dotfiles
mkdir: 无法创建目录“/home/lxz/glts/dotfiles”: 没有那个文件或目录
lxz@lxz-virtual-machine:~/文档/system-development-tool$ sleep 10000
^Z
[1]+  已停止                  sleep 10000
lxz@lxz-virtual-machine:~/文档/system-development-tool$ bg
^Z
LibreOffice Calc  -machine:~/文档/system-development-tool$
^Z
lxz@lxz-virtual-machine:~/文档/system-development-tool$ sleep
52225
lxz@lxz-virtual-machine:~/文档/system-development-tool$ pgrep sleep
pgrep: only one pattern can be provided
try 'pgrep --help' for more information.
lxz@lxz-virtual-machine:~/文档/system-development-tool$ pkill -af sleep
pkill: 无效选项 -- a
Usage:
pkill [options] <pattern>
Options:
  -s, --signal <sig>    signal to send (either number or name)
  -e, --echo             display what is killed
  -c, --count            count of matching processes
  -f, --full             use full process name to match
  -n, --porcupine <PID> match listed process group IDs
```

图 2: Caption

3. 使用 `sleep 60 &` 作为先执行的程序。然后使用 `wait` 命令。尝试启动这个休眠命令，然后待其结束后再执行 `ls` 命令

```
lxz@lxz-virtual-machine:~/文档/system-development-tool$ sleep 60 &
[1] 15357
lxz@lxz-virtual-machine:~/文档/system-development-tool$ pgrep sleep | wait
lxz@lxz-virtual-machine:~/文档/system-development-tool$ ls
buggy.sh  html_root  out.log  vimrc  words  第一次实验报告.pdf
for.sh    narco.sh  until.sh  while.sh  第二次实验报告.pdf
```

图 3: Caption

4. 使用 `fg` 或 `bg` 命令恢复暂停的工作。它们分别表示在前台继续或在后台继续。

```
lxz@lxz-virtual-machine:~/文档/System-Development-Tool-$ /bin/sleep 100 &
[1] 3067
lxz@lxz-virtual-machine:~/文档/System-Development-Tool-$ jobs
[1]+  运行中                  /bin/sleep 100 &
lxz@lxz-virtual-machine:~/文档/System-Development-Tool-$ fg %1
/bin/sleep 100
```

图 4: Caption

5. 为了防止一旦关闭终端(会发送另外一个信号 `SIGHUP`), 这些后台的进程也会终止, 可以使用 `nohup` (一个用来忽略 `SIGHUP` 的封装) 来运行程序

```
sleep 1000
lxz@lxz-virtual-machine:~/文档/System-Development-Tool-$ nohup sleep 2000 &
[3] 3113
lxz@lxz-virtual-machine:~/文档/System-Development-Tool-$ nohup: 忽略输入并把输出追加到 'nohup.out'
lxz@lxz-virtual-machine:~/文档/System-Development-Tool-$ jobs
[2]+  已停止                  sleep 1000
[3]-  运行中                  nohup sleep 2000 &
lxz@lxz-virtual-machine:~/文档/System-Development-Tool-$
```

图 5: Caption

## 6. tmux 中对象的继承结构之对话相关

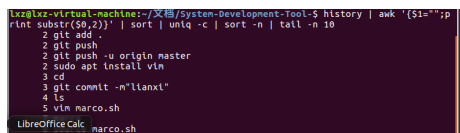
1. tmux 开始一个新的会话
  2. `tmux new -s NAME` 以指定名称开始一个新的会话
  3. `tmux ls` 列出当前所有会话
  4. 在 `tmux` 中输入 `<C-b> d`, 将当前会话分离
  5. `tmux a` 重新连接最后一个会话。您也可以通过 `-t` 来指定具体的会话
- 相关实验截图:

```
lxz@lxz-virtual-machine:~/文档/System-Development-Tool-$ tmux ls
0: 1 windows (created Tue Sep 10 19:55:25 2024) [80x23]
lxz: 1 windows (created Tue Sep 10 19:58:05 2024) [80x23] (attached)
lxz@lxz-virtual-machine:~/文档/System-Development-Tool-$
```

图 6: Caption

## 7. 长串包含许多选项的命令会非常麻烦。因此，大多数 shell 都支持设置别名

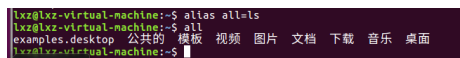
先查找常用的指令



```
lxz@lxz-virtual-machine:~/文档/System-Development-Tool$ history | awk '{ $1="" } ; p
rint substr($0,20)' | sort | uniq -c | sort -n | tail -n 10
  2 git add .
  2 git push
  2 git push -u origin master
  2 sudo apt install vim
  3 cd
  3 git commit -m "lanxi"
  4 ls
  5 vim marco.sh
```

图 7: Caption

然后其中 ls 的别名设置为 all



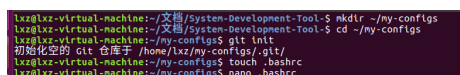
```
lxz@lxz-virtual-machine:~$ alias all=ls
lxz@lxz-virtual-machine:~$ all
examples.desktop 公共的 模板 视频 图片 文档 下载 音乐 桌面
lxz@lxz-virtual-machine:~$
```

图 8: Caption

## 8. 为配置文件创建一个新文件夹，并设置版本控制，同时添加一个自定义的 shell 配置文件。

1. 创建一个新的文件夹（my-configs）
2. 进入刚刚创建的文件夹并初始化 Git 仓库
3. 在文件夹内创建一个新的配置文件，比如说 bashrc
4. 添加自定义设置

相关实验截图



```
lxz@lxz-virtual-machine:~/文档/System-Development-Tool$ mkdir ~/my-configs
lxz@lxz-virtual-machine:~/文档/System-Development-Tool$ cd ~/my-configs
lxz@lxz-virtual-machine:~/my-configs$ git init
初始化空的 git 仓库于 /home/lxz/my-configs/.git/
lxz@lxz-virtual-machine:~/my-configs$ touch .bashrc
lxz@lxz-virtual-machine:~/my-configs$ nano .bashrc
```

图 9: Caption

## 9. 将新的配置文件添加到 git 仓库并提交更改

相关实验截图

```
lxz@lxz-virtual-machine:~/my-configs$ nano .bashrc
lxz@lxz-virtual-machine:~/my-configs$ git add .bashrc
lxz@lxz-virtual-machine:~/my-configs$ git commit -m "Add initial .bashrc with custom settings"
(master) 87468dc Add initial .bashrc with custom settings
LibreOfficeWriter 4 insertions(+)
Create mode 100644 .bashrc
lxz@lxz-virtual-machine:~/my-configs$ git remote add origin https://github.com/l
```

图 10: Caption

## 10. 将已有的远程仓库添加为远程并将本地提交推送到远程仓库

相关实验截图

```
lxz@lxz-virtual-machine:~/my-configs$ git remote add origin https://github.com/l
LibreOfficeWriter development-Tool-.git
lxz@lxz-virtual-machine:~/my-configs$ git push --set-upstream origin master
Username for 'https://github.com': lxiezhen
Password for 'https://lxiezhen@github.com':
对象计数中: 3, 完成。
LibreOfficeCalc 3 (3/3), 271 bytes | 0 bytes/s, 完成。
remote: Create a pull request for 'master' on GitHub by visiting:
remote: https://github.com/lxiezhen/System-Development-Tool-/pull/new/master
To https://github.com: master -> master
分支 master 设置为跟踪来自 origin 的远程分支 master。
lxz@lxz-virtual-machine:~/my-configs$
```

图 11: Caption

## 11. 使用 ssh-keygen -o -a 100 -t ed25519 来创建一个密钥

相关实验截图

```
lxz@lxz-virtual-machine:~/my-configs$ ssh-keygen -o -a 100 -t ed25519 -f ~/.ssh/
id_ed25519
Generating public/private ed25519 key pair.
/home/lxz/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/lxz/.ssh/id_ed25519.
Your public key has been saved in /home/lxz/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:/LAC/7WQXoIZ3ZYfM8SB3XlYkd0nM4IasyhaHo+uP3eo lxz@lxz-virtual-machine
The key's randomart image is:
+--[ED25519 256]--+
| . . . . .B* |
| + 0 0 . +00+ |
| 0 = . 0 +0000 |
| . + . . . .0 |
| 0 . S + . . |
| . . 0 . . . |
| . . 0 * . . |
| . + . . +0 |
| 0E.. . +0 |
+----[SHA256]-----+
```

图 12: Caption

## 12. 修改 config 中的配置

相关实验截图

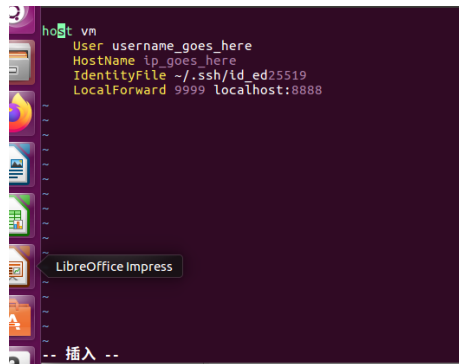


图 13: Caption

### 13. 将 ssh 密钥拷贝到服务器

相关实验截图

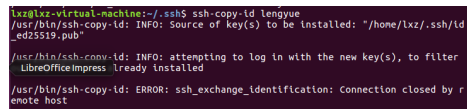


图 14: Caption

### 14. 使用 `python -m http.server 8888` 在您的虚拟机中启动一个 Web 服务器并通过本机的 `http://localhost:9999` 访问虚拟机上的 Web 服务器

相关实验截图

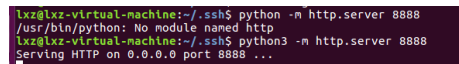


图 15: Caption

### 15. 编辑 SSH 服务器配置，禁用密码验证

相关实验截图

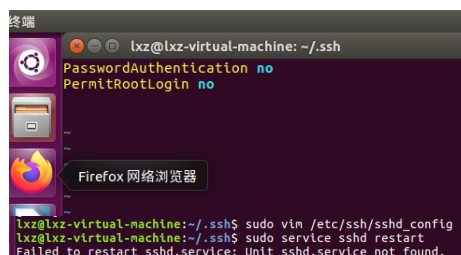


图 16: Caption

python 视觉应用相关实验:

## 16. 对载入的图像实现灰度变换

相关实验截图

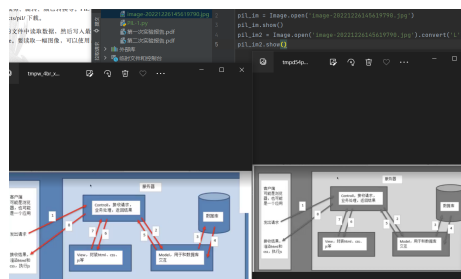


图 17: Caption

## 17. 使用 PIL 创建图像的缩略图

相关实验截图

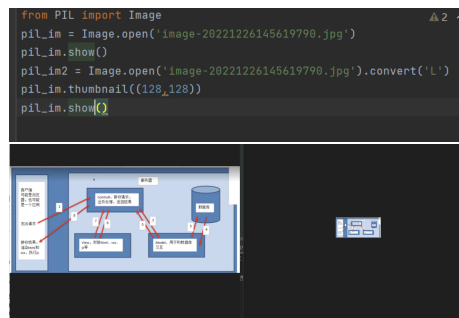


图 18: Caption

## 18. 复制和粘贴图像区域

相关实验截图

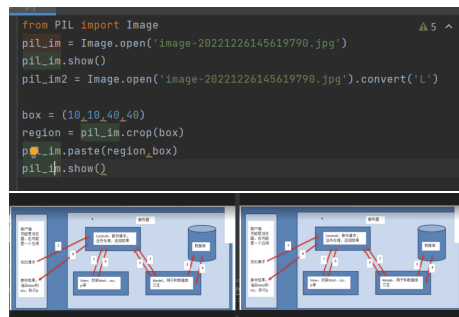


图 19: Caption

## 19. 调整尺寸和旋转

相关实验截图



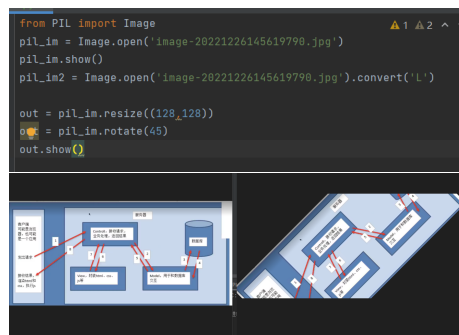


图 20: Caption

## 20. 绘制图像、点和线

相关实验截图

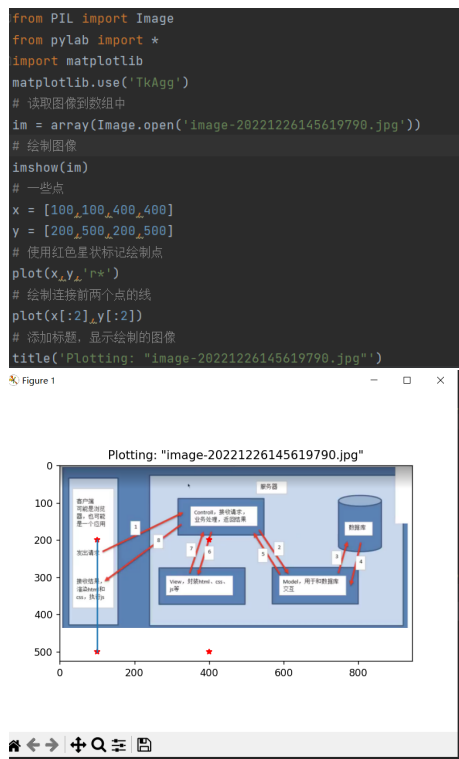


图 21: Caption

## 四. 实验中出现的問題以及解决办法

### 1. 解释器执行不对

在虚拟机中启动一个 Web 服务时使用 `python -m http.server 8888` 语句一支不对，后面才反应过来自己 `python` 的解释器时 `python3`，改后成功执行

```
lxz@lxz-virtual-machine:~/.ssh$ python -m http.server 8888
/usr/bin/python: No module named http
lxz@lxz-virtual-machine:~/.ssh$ python3 -m http.server 8888
Serving HTTP on 0.0.0.0 port 8888 ...
```

图 22: Caption

### 2. 绘制图像、点和线时代码执行错误

在最初执行语句时代码报错，寻找网络相关问题后通过加两行代码解决的

```
canvas_class = cast(type(FigureCanvasBase), _get_backend_mod().FigureCanvas)
# canvas_class = module.FigureCanvas
import matplotlib
matplotlib.use('TkAgg')
```

图 23: Caption

## 五. 实验收获与心得

本次的实验学习了命令行环境，Python 入门基础，Python 视觉应用三个大模块。其中特别详细的学习了命令行和 `python` 的相关用法。其中命令行环境的学习让我对任务控制，进程管理等有了一定的了解。还接触到了 `tmux` 这个新鲜，以前未成接触过的东西。接着在 `python` 的学习中，让我学会了有关图像的处理。总之这一次的实验知识很丰富，让我也收获很多，我相信这对我之后的实验都是有所帮助的。

仓库链接：<https://github.com/lixiezhn/System-Development-Tool->