

实验报告

陈俊泽

August 2024

目录

1 练习实例	1
2 解题感悟	3

1 练习实例

课后题部分部分

任务控制部分

实例 1（第一题）：使用 `pgrep sleep` 来找到之前在终端执行的 `sleep 10000` 任务，使用 `pkill -f sleep` 即可直接结束进程

实例 2（第二题）：先用 `sleep 60` 命令来开始休眠程序，接着使用 `pgrep` 来查看休眠程序的 `pid` 号，以 `pid` 号来作为 `pidwait` 的入口参数，`pidwait` 代码如下 `while kill -0 $1`

`do`

`sleep 1`

`done`

`ls`

终端多路复用部分

实例 3（第一题）：使用命令 `alias dc=cd` 即可为 `cd` 创建别名 `dc`

实例 4（第二题）：执行 `history | awk '{1 = ""; print substr(0,2)}' | sort | uniq -c | sort -n | tail -n 10` 后根据输出，再利用 `alias` 命令 = 别名即可

配置文件部分

实例 5(第一题): 创建文件夹的命令为 `mkdir /config_files`, 版本控制 `git init`

实例 6 (第二题): 先创建 `.bashrc` 文件: `nano /.bashrc`, 自定义提示符:
`export PS1='[@]$`

实例 7 (第三题): 先创建脚本文件: `nano /config_files/setup.sh`, 接着编写内容: `#!/bin/bash # 切换到配置文件夹 CONFIG_DIR= /config_files`
`# 软链接配置文件到用户主目录`

`ln -sf $CONFIG_DIR/.bashrc /.bashrc`

`ln -sf $CONFIG_DIR/.zshrc /.zshrc`

`echo " 配置文件安装完毕"`

实例 8 (第四题): 安装脚本: `/config_files/setup.sh`

远端设备部分

实例 9: (第一题): 首先用 `ssh-keygen` 命令生成密钥, 并利用 `ssh-copy-id vm` 命令将自己的公钥送到服务器中

实例 10 (第二题): 修改 `sshd_config` 中的配置文件, 将 `no` 改成 `yes`, 使用 `sudo service sshd restart` 重启 `ssh` 服务器即可

实例 11 (第三题): 用 `sudo apt-get install mosh` 命令下载 `mosh`, 并用 `mosh-server` 命令启动连接, 用 `sudo apt install mosh` 和 `mosh my_server@`, 实现使用 `mosh` 连接虚拟机, 最后结果表明不可以恢复连接

实例 12(附加题): 通过查看, `N` 是不执行远端命令, `f` 是让 `ssh` 在执行命令前切换到后台运行后台进行端口转发具体实现命令为 `ssh -fN -L 9999:localhost:8888 pi`

python 部分

实例 13: 调整图像的大小是调用 `resize` 方法, 输入的参数是一个二维组, 用于决定图片的大小

实例 14: 调整图像的方向使用 `rotate` 方法, 输入的参数是一个旋转的角度

实例 15: 复制图像用到 `crop` 方法, 首先先用四元组确定复制粘贴的区域再用 `crop ()` 即可

实例 16: 图像数组化用到的是 `NumPy` 工具包中的 `array` 方法, 正常来说, 转换后图像有三维, 为行、列、颜色通道, 并且是 `uint8` 类型, 也可以指定去灰度化, 即去除颜色通道和等等一系列操作

实例 17: 图片的灰度变化主要是通过 0...255 区间的映射变化来进行, 如

```
im2 = 255 - im # 对图像进行反相处理
```

```
im3 = (100.0/255) * im + 100 # 将图像像素值变换到 100...200 区间
```

```
im4 = 255.0 * (im/255.0)**2 # 对图像像素值求平方后得到的图像
```

都可以进行灰度处理

实例 18: 直方图均衡化, 即将图像的灰度直方图变平, 使变换后的图像中每个灰度值的分布概率都相同, 它是这样实现的: 首先, 统计图像中每个灰度值出现的频率, 得到图像的直方图, 将直方图中的频率累加, 得到 CDF 函数。将 CDF 归一化, 使其范围在 $[0, 255]$ 之间, 使用 CDF 对图像的每个像素进行重映射, 将映射后的像素值重新组合成图像, 得到对比度增强后的结果即可。

实例 19: 图像平均是一种处理图像噪声和创建艺术效果的方法, 首先要将图像存储为浮点型数组, 然后将图像列表中的所有图像, 将每幅图像的像素值逐一加到当前的平均图像上, 将累加后的结果除以图像的数量, 从而得到平均图像, 最后将平均图像的数据类型转换回 `uint8` 即可

实例 20: `pickle` 模块可以实现对数据的封装和和拆分。使用 `pickle.dump(obj,file)` 将对象 `obj` 写入到文件 `file` 中, 使用 `pickle.load(file)` 从文件 `file` 中读取对象。

2 解题感悟

通过这次试验, 我学习到了如何有效管理进程。如使用 `pgrep` 命令能够方便地找到进程 ID, 而 `pkill` 命令则让我们可以根据进程名称或模式直接终止进程。同时还学习了终端多路复用部分主要涉及到命令别名的创建和管理以及如何创建和管理配置文件夹, 使用 `git` 进行版本控制, 以及编写脚本来管理配置文件。除此之外, 我还学习到如何利用密钥和 `ssh` 方法来实现免密远程连接, 最后学习了一些计算机视觉中利用 `python` 来处理图像的一些方法。