

实验报告

程传哲

2024 年 9 月 11 日

1 学习成果

1.1 命令行环境

1. 中断执行任务结束进程

- shell会使用unix提供的信号机制执行进程间的通信
我们可以使用CTRL-C来停止任务的进程
但有时候SIGNIT信号被捕获无法停止进程
这个时候可以通过CTRL-/实现程序进程中断

2. 终端执行程序暂停与后台执行

- SIGSTOP信号可以使进程暂停，在Linux shell中输入CTRL-Z可以发送该信号
使用fg与bg命令可以回复暂停的工作任务，前台进行或后台进行

3. tmux终端多路复用器

- 终端多路复用器分割窗口
- tmux多线路有多种指令
 - - tmux开始新会话
 - - tmux new -s name以一个固定名字开始一个会话

- `tmux ls` 列出当前所有会话
- `tmux -a` 连接到最后一个会话 `-t`或指定会话
- `-<C-b >c` 创建一个新的窗口
- `-<C-b >N` 跳转到第N个窗口
- `-<C-b >p` 切换到前一个窗口
- `-<C-b >n` 切换到下一个窗口
- `-<C-b >w` 列出当前所有窗口

4. 设置别名

- alias的使用
 - `alias la = "ls -a"`
 - `unalias la` 禁用la
 - `alias ll` 获取别名定义

5. Dotfiles配置文件

- 对于很对隐藏文件他们不会被ls显示因为他们的开头以点命名
- 对于bash来说可以通过编辑`bashrc`或是`bash _profile`来进行配置
- 很多情况下需要包含PATH的一些指令以便这些程序被shell找到

6. SSH远端设备

- 连接到远端服务器
 - 通过URL指定登录`ssh name@xxxx.com`
 - 痛过ip地址指定登录`name@xxx.xxx.x.xx`

7. 生成ssh密钥

- `ssh-keygen -o -a 100 -t ed25519 -f ~/.ssh/id_ed25519`
- `ssh-keygen -t rsa -C "****@gmail.com"`通过这些指令可以生成一对密钥分别为私钥与公钥

8. ssh复制文件

- `scp filename username@ip_address /home/username` 但scp现在已经被弃用
- `rsync username@ip_address /home/username/filename .` 使用rsync不仅可以复制文件并且可以复制整个目录
- `ssh + tee cat localfile — ssh remote_server tee serverfile` 将其标准输出到另一个文件

9. ssh配置

- 设置别名: `alias my_server= ssh -i ~/id_ed25519 -port 2222 -L 9999 localhost 8888 foobar@remote_server`
- config设置
 - Host vm
 - * User foobar
 - * HostName 172.16.174.141
 - * Port 2222
 - * IdentityFile ~/ssh/id_ed25519
 - * LocalForward 9999 localhost 8888

1.2 python的基本使用

1. print

- print打印语句

2. if else 语句

```

1      num = 5
2      if num == 3:
3          print 'boss'
4      elif num == 2:
5          print 'user'
6      elif num == 1:
7          print 'worker'
8      elif num < 0:
9          print 'error'
10     else:
11         print 'roadman'

```

3. while 语句

```

1      count = 0
2      while (count < 9):
3          print 'The count is:', count
4          count = count + 1
5      print "Good bye!"

```

4. for 循环

```

1      for num in range(10,20):
2          for i in range(2,num):
3              if num%i == 0:
4                  j=num/i
5                  print ('%d = %d * %d' % (num,i,j))
6                  break
7          else:
8              print ('%d is primer number' % num)

```

5. continue语句

```

1      for letter in 'Python':

```

```

2         if letter == 'h':
3             continue
4         print 'now :', letter

```

6. pass语句

- python pass语句是空语句，是为了保持程序结构的完整性

```

1     for letter in 'Python':
2         if letter == 'h':
3             pass
4             print 'this is pass'
5         print 'now letter :', letter

```

7. 列表

- 序列都可以进行的操作包括索引，切片，加，乘，检查成员

```

1 list01 = ['runoob', 786, 2.23, 'john', 70.2]
2 list02 = [123, 'john']
3
4 print list01
5 print list02
6
7 print list01[0]
8 print list01[-1]
9 print list01[0:3]
10
11
12 print list01 * 2
13
14
15 print list01 + list02
16
17 print len(list01)

```

```
18
19
20 del list02[0]
21 print list02
22
23
24 print 'john' in list02 # True
25
26
27 for i in list01:
28     print i
29
30
31 print cmp(list01, list02)
32
33
34 print max([0, 1, 2, 3, 4])
35 print min([0, 1])
36
37
38 aTuple = (1,2,3,4)
39 list03 = list(aTuple)
40 print list03
41
42
43 list03.append(5)
44 print list03
45
46
47 list03.extend(list01)
48 print list03
49
50
```

```
51 print list03.count(1)
52
53
54 print list03.index('john')
55
56
57 list03.insert(0, 'hello')
58 print list03
59
60
61 print list03.pop(0)
62 print list03
63
64
65 list03.remove(1)
66 print list03
67
68
69 list03.reverse()
70 print list03
71
72
73 list03.sort()
74 print list03
```

8. 模块

- 导入模块import support等

9. 函数

```
1 def functionname( parameters ) :
2     function_suite
3     return [expression]
```

1.3 pyhton视觉

1. 基本画图

```
1 from PIL import Image
2 from pylab import *
3
4 im = array(Image.open('1.jpg').convert('L'))
5
6 x = [100, 200, 300, 400]
7 y = [100, 300, 400, 200]
8 plot(x, y, 'r*')
9 plot(x[:2], y[:2])
10 title('yuandonglin')
```

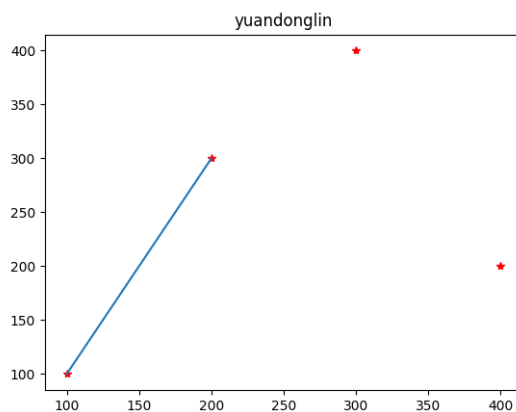


图 1: figure title

2. 处理图像

```
1 from PIL import Image
2 from pylab import *
3
4 im = array(Image.open('1.jpg').convert('L'))
5
6 figure()
```



```
7 contour(im, origin='image')
8 axis('equal')
9 axis('off')
10 figure()
11 hist(im.flatten(), 128)
12 show()
```

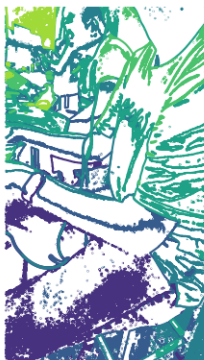


图 2: figure title

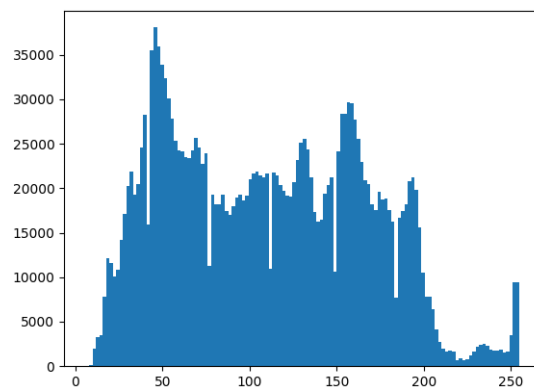


图 3: figure title

3. 输入读取语句

- `read num` 回车即可操作
- `read -p "请输入num的值" num` 其中加入-p可以同时显示与读取

4. 只读变量

- `readonly num = 10`回车即可操作后续将无法通过赋值语句改变num的量

2 Github链接

<https://github.com/Chengchuanzhe/-.git>

3 学习感悟

在本次的系统工具开发实践课中我受益颇多，在深入学习Shell与Vim的过程中，我经历了从初识的困惑到逐渐熟练运用的转变，这一过程不仅提升了我的技术能力，也让我对编程和命令行操作有了更深的理解和感悟，在学习shell中逐渐的了解了各个指令的用处以及shell脚本的使用，在vim的学习中学会了如何去使用vim键盘以及编写程序，这次的课程使我收益匪浅，也会继续在系统操作中继续学习下去