《Linux 系统应用与开发》

期末报告

课程名称　　Linux 系统应用与开发

题 目　　期末课程设计报告

学生姓名　　常文瀚

学生学号　　20181001095

学生班级　　191181

日 期 ：　　2020 年 12 月 20 日

目录

一、安装 Tensorflow，并通过 MNIST 训练模型对图像进行识别

1.设计内容

基于 Tensorflow 框架与 MNIST 数据集可以识别手写体的神经网络模型，并将模型保存下来，实现对单张图片的分类任务

2.具体设计

（1） 设计思路

① 在阿里云轻量服务器中安装 Tensorflow 框架；

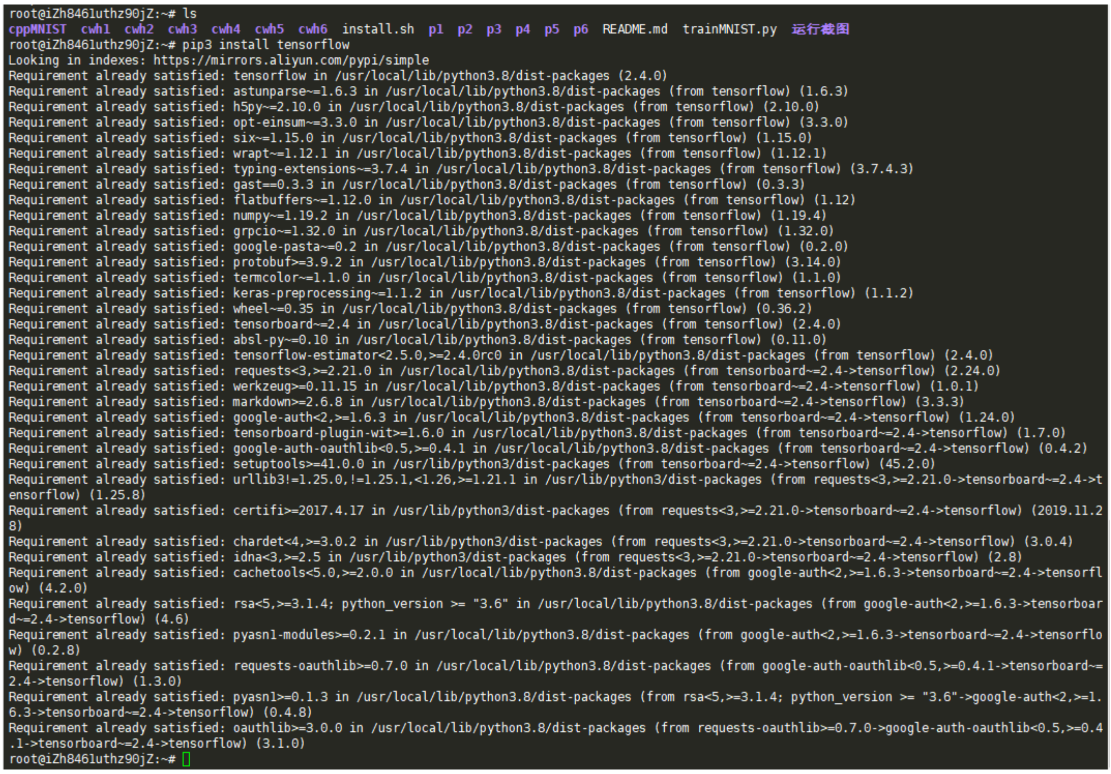
② 在服务器中安装 Jupyter Notebook 以实现识别单张图片的功能；

③ 运行 Jupyter Notebook 并运行代码，保存训练好的模型；

④ 调用模型，对图像进行识别。

（2） 操作步骤

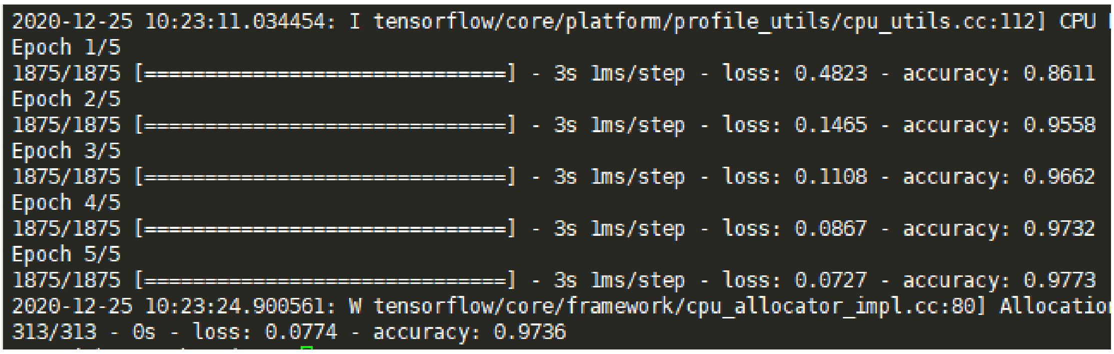
① 安装 Tensorflow



```
root@iZh8461uthz90jZ:~# ls
cppMNIST  cwh1  cwh2  cwh3  cwh4  cwh5  cwh6  install.sh  p1  p2  p3  p4  p5  p6  README.md  trainMNIST.py  运行截图
root@iZh8461uthz90jZ:~# pip3 install tensorflow
Looking in indexes: https://mirrors.aliyun.com/pypi/simple
Requirement already satisfied: tensorflow in /usr/local/lib/python3.8/dist-packages (2.4.0)
Requirement already satisfied: astunparse~=1.6.3 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: h5py~=2.10.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (2.10.0)
Requirement already satisfied: opt-einsum~=3.3.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: six~=1.15.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (1.15.0)
Requirement already satisfied: wrapt~=1.12.1 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (1.12.1)
Requirement already satisfied: typing-extensions~=3.7.4 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (3.7.4.3)
Requirement already satisfied: gast==0.3.3 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (0.3.3)
Requirement already satisfied: flatbuffers~=1.12.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (1.12)
Requirement already satisfied: numpy~=1.19.2 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (1.19.4)
Requirement already satisfied: grpcio~=1.32.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (1.32.0)
Requirement already satisfied: google-pasta~=0.2 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (3.14.0)
Requirement already satisfied: termcolor~=1.1.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (1.1.0)
Requirement already satisfied: keras-preprocessing~=1.1.2 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (1.1.2)
Requirement already satisfied: wheel~=0.35 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (0.36.2)
Requirement already satisfied: tensorboard~=2.4 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (2.4.0)
Requirement already satisfied: absl-py~=0.10 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (0.11.0)
Requirement already satisfied: tensorflow-estimator<2.5.0,>=2.4.0rc0 in /usr/local/lib/python3.8/dist-packages (from tensorflow) (2.4.0)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.8/dist-packages (from tensorboard~=2.4->tensorflow) (2.24.0)
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.8/dist-packages (from tensorboard~=2.4->tensorflow) (1.0.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.8/dist-packages (from tensorboard~=2.4->tensorflow) (3.3.3)
Requirement already satisfied: google-auth<2,>=1.6.3 in /usr/local/lib/python3.8/dist-packages (from tensorboard~=2.4->tensorflow) (1.24.0)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.8/dist-packages (from tensorboard~=2.4->tensorflow) (1.7.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.8/dist-packages (from tensorboard~=2.4->tensorflow) (0.4.2)
Requirement already satisfied: setuptools>=41.0.0 in /usr/lib/python3/dist-packages (from tensorboard~=2.4->tensorflow) (45.2.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/lib/python3/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.4->t
ensorflow) (1.25.8)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.4->tensorflow) (2019.11.2
8)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/lib/python3/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.4->tensorflow) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/lib/python3/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.4->tensorflow) (2.8)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from google-auth<2,>=1.6.3->tensorboard~=2.4->tensorfl
ow) (4.2.0)
Requirement already satisfied: rsa<5,>=3.1.4; python_version >= "3.6" in /usr/local/lib/python3.8/dist-packages (from google-auth<2,>=1.6.3->tensorboar
d~=2.4->tensorflow) (4.6)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.8/dist-packages (from google-auth<2,>=1.6.3->tensorboard~=2.4->tensorflo
w) (0.2.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.8/dist-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=
2.4->tensorflow) (1.3.0)
Requirement already satisfied: pyasn1>=0.1.3 in /usr/local/lib/python3.8/dist-packages (from rsa<5,>=3.1.4; python_version >= "3.6"->google-auth<2,>=1.
6.3->tensorboard~=2.4->tensorflow) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.8/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4
.1->tensorboard~=2.4->tensorflow) (3.1.0)
root@iZh8461uthz90jZ:~# 
```

图 1 安装 Tensorflow 成功

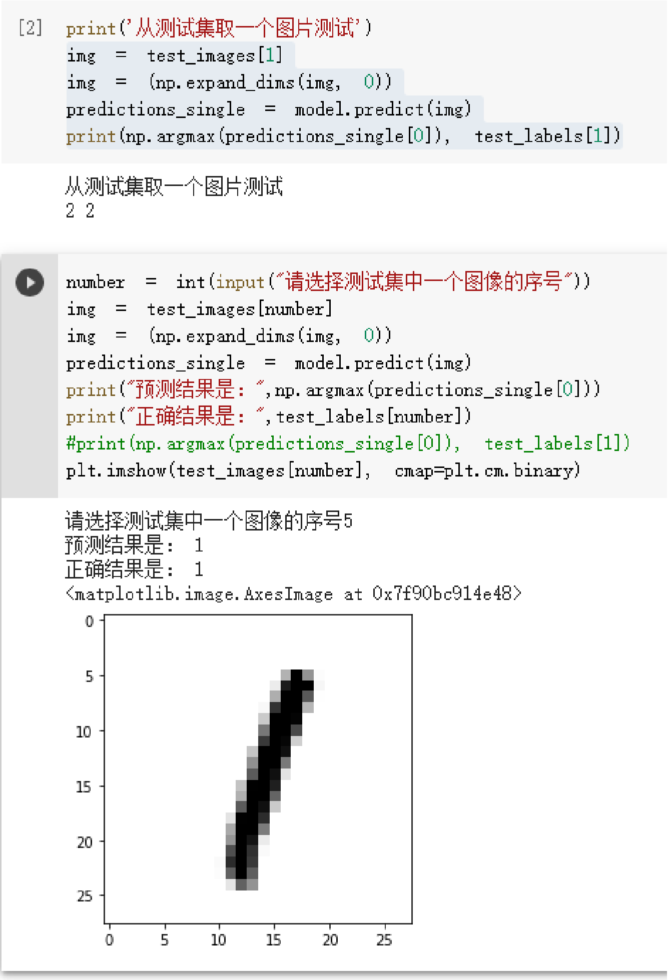
② 进行训练



```
2020-12-25 10:23:11.034454: I tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU
Epoch 1/5
1875/1875 [==============================] - 3s 1ms/step - loss: 0.4823 - accuracy: 0.8611
Epoch 2/5
1875/1875 [==============================] - 3s 1ms/step - loss: 0.1465 - accuracy: 0.9558
Epoch 3/5
1875/1875 [==============================] - 3s 1ms/step - loss: 0.1108 - accuracy: 0.9662
Epoch 4/5
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0867 - accuracy: 0.9732
Epoch 5/5
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0727 - accuracy: 0.9773
2020-12-25 10:23:24.900561: W tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocatio
313/313 - 0s - loss: 0.0774 - accuracy: 0.9736
```

图 2 训练模型结束并得出模型

③ 调用模型测试



```
[2]  print('从测试集取一个图片测试')
     img  =  test_images[1]
     img  =  (np.expand_dims(img,  0))
     predictions_single  =  model.predict(img)
     print(np.argmax(predictions_single[0]),  test_labels[1])
```

```
从测试集取一个图片测试
2 2
```

```
number  =  int(input("请选择测试集中一个图像的序号"))
img  =  test_images[number]
img  =  (np.expand_dims(img,  0))
predictions_single  =  model.predict(img)
print("预测结果是：",np.argmax(predictions_single[0]))
print("正确结果是：",test_labels[number])
#print(np.argmax(predictions_single[0]),  test_labels[1])
plt.imshow(test_images[number],  cmap=plt.cm.binary)
```

```
请选择测试集中一个图像的序号5
预测结果是：  1
正确结果是：  1
<matplotlib.image.AxesImage at 0x7f90bc914e48>
```

（3） 具体代码

from __future__ import absolute_import, division, print_function, unicode_literals

import matplotlib.pyplot as plt

import tensorflow as tf

```
import numpy as np

import os


mnist = tf.keras.datasets.mnist


(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

train_images, test_images = train_images / 255.0, test_images / 255.0


#  显示一部分图片数据

# plt.figure(figsize=(10, 10))

# class_names = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']

# for i in range(25):

#       plt.subplot(5, 5, i + 1)

#       plt.xticks([])

#       plt.yticks([])

#       plt.grid(False)

#       plt.imshow(train_images[i], cmap=plt.cm.binary)

#       plt.xlabel(class_names[train_labels[i]])

# plt.show()


if os.path.exists('./model.h5'):

    model = tf.keras.models.load_model('./model.h5')

else:

    model = tf.keras.models.Sequential([

        tf.keras.layers.Flatten(input_shape=(28, 28)),

        tf.keras.layers.Dense(128, activation='relu'),

        tf.keras.layers.Dropout(0.2),

        tf.keras.layers.Dense(10, activation='softmax')

    ])

    #
```

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

model.fit(train_images, train_labels, epochs=5)

model.save('model.h5')

print('在测试集上评估')

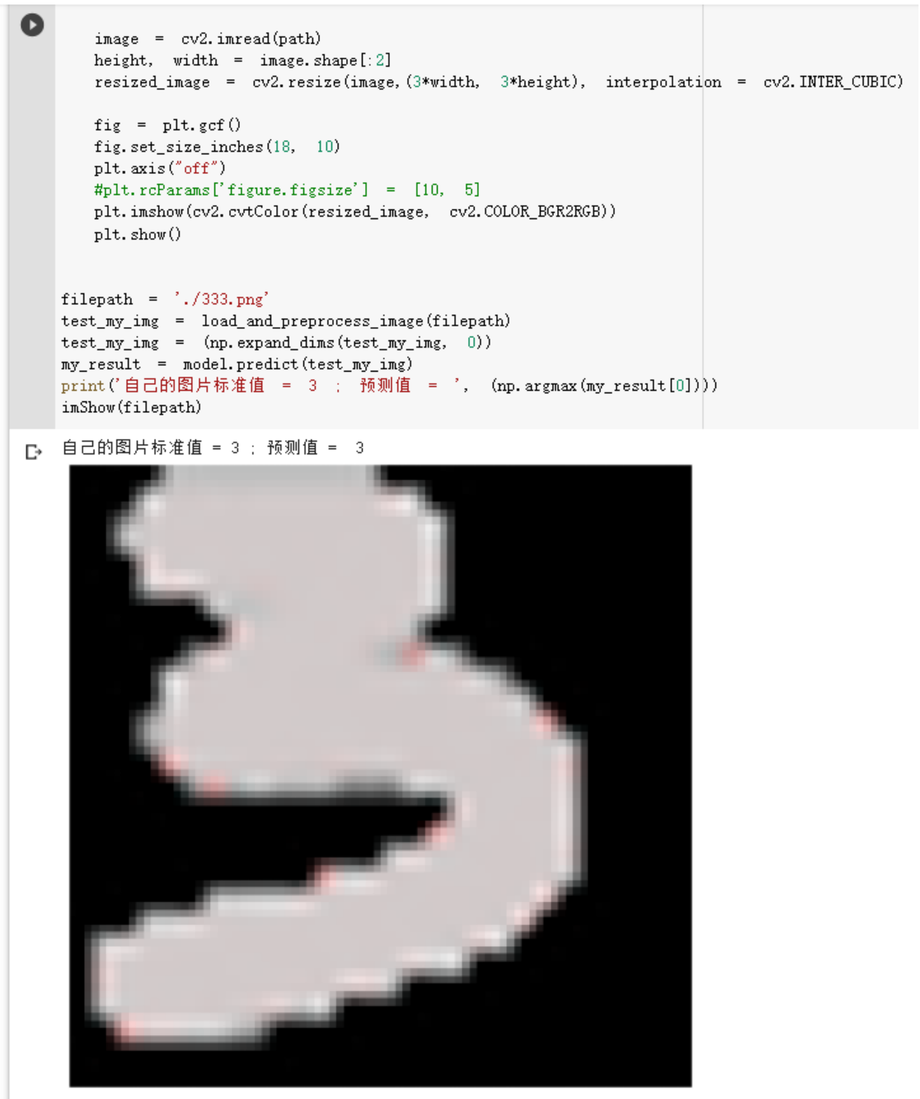
test_loss, test_acc = model.evaluate(test_images, test_labels, verbose=2)

print('看看测试集测试结果')

predictions = model.predict(test_images)

print('预测值 = %i ; 正确值 = %i' % (np.argmax(predictions[0]), test_labels[0]))

（4） 运行结果



```
    image = cv2.imread(path)
    height, width = image.shape[:2]
    resized_image = cv2.resize(image, (3*width, 3*height), interpolation = cv2.INTER_CUBIC)

    fig = plt.gcf()
    fig.set_size_inches(18, 10)
    plt.axis("off")
    #plt.rcParams['figure.figsize'] = [10, 5]
    plt.imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))
    plt.show()


filepath = './333.png'
test_my_img = load_and_preprocess_image(filepath)
test_my_img = (np.expand_dims(test_my_img, 0))
my_result = model.predict(test_my_img)
print('自己的图片标准值 = 3 ; 预测值 = ', (np.argmax(my_result[0])))
imShow(filepath)
```

```
自己的图片标准值 = 3 ; 预测值 =  3
```

3.总结

通过课上学习的知识，我初步学习到了 Linux 命令行操作的方法，但是在上机时我却遇到了一些困难，例如在使用 python 命令时，需要写入 python3 而不是 python，否则将使用的是 python2 版本，在经过一系列的资料查询后，我能够顺利的使用命令行进行操作，第一次的实验使我受益非浅。

二、进行命令行操作，安装 Nginx，对搜狗数据集进行操作

1.设计内容

（1）对老师给出的上级任务文件中的命令进行实践

（2）安装 Nginx 并进行访问

（3）下载搜狗数据集，并进行对应的数据操作

了解数据集作用及格式（本实验下载精简版即可,一般是几十 K)。

在 linux 中解压缩并用相关文本命令进行数据查看、统计及分析。(包括但不限于：统计某关键词的词频、对文件进行排序（按照某列或某关键字)，对大文件的快速浏览，管道命令、模糊查询（使用通配符）等)。

2.具体设计

（1） 设计思路

① 先对给出的命令进行实践操作

② 下载 Nginx，在安装后访问 Linux 系统的 IP 地址验证是否成功

③ 下载搜狗数据集，在 linux 中解压缩并用相关文本命令进行数据查看、统计及分析。（包括但不限于：统计某关键词的词频、对文件进行排序（按照某列或某关键字)，对大文件的快速浏览，管道命令、模糊查询（使用通配符）等)。

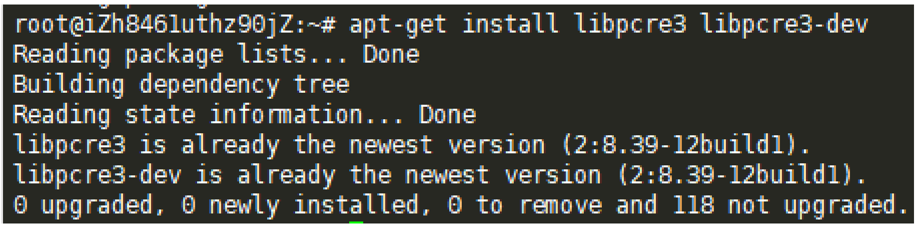
（2） 操作步骤

① 根据第二次上机任务中的指令进行操作；

② 执行同步获取最新软件包；

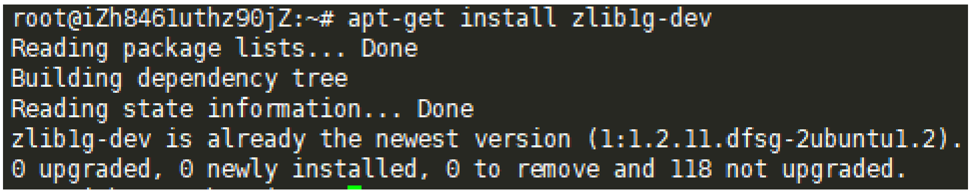
③ 安装 PCRE，PCRE(Perl Compatible Regular Expressions)是一个 Perl 库，包括 perl 兼容的正则表达式库。nginx 的 http 模块使用 pcre 来解析正则表达式，pcre-devel 是使用 pcre

开发的一个二次开发库。nginx 也需要此库；



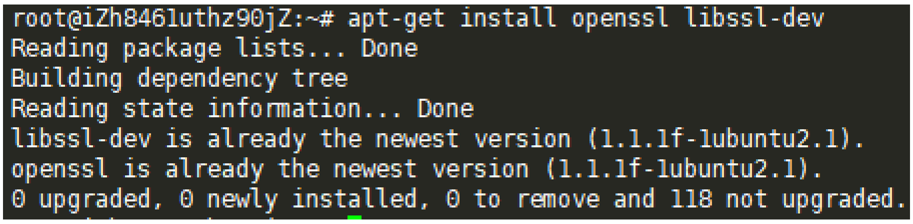
```
root@iZh846luthz90jZ:~# apt-get install libpcre3 libpcre3-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libpcre3 is already the newest version (2:8.39-12build1).
libpcre3-dev is already the newest version (2:8.39-12build1).
0 upgraded, 0 newly installed, 0 to remove and 118 not upgraded.
```

④ 安装 zlib，zlib 库提供了很多种压缩和解压缩的方式，nginx 使用 zlib 对 http 包的内容进行 gzip；



```
root@iZh846luthz90jZ:~# apt-get install zlib1g-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
zlib1g-dev is already the newest version (1:1.2.11.dfsg-2ubuntu1.2).
0 upgraded, 0 newly installed, 0 to remove and 118 not upgraded.
```

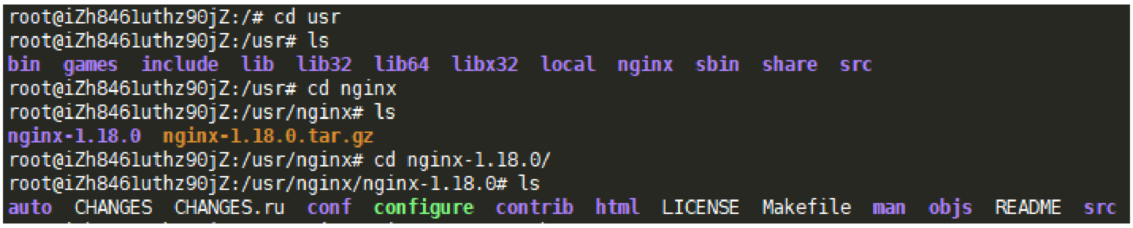
⑤ 安装 OpenSSL ，OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及 SSL 协议，并提供丰富的应用程序供测试或其它目的使用，nginx 不仅支持 http 协议，还支持 https（即在 ssl 协议上传输 http）；



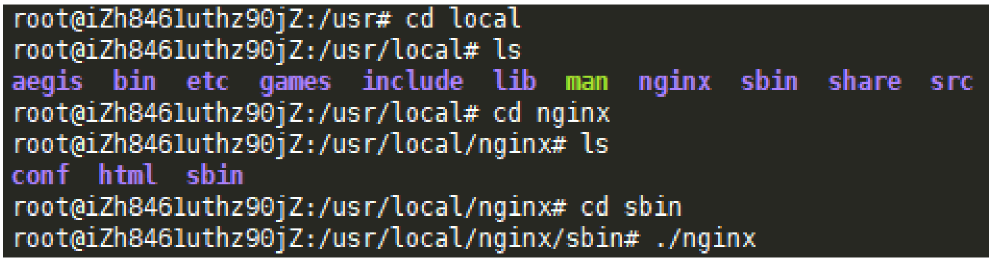
```
root@iZh846luthz90jZ:~# apt-get install openssl libssl-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libssl-dev is already the newest version (1.1.1f-1ubuntu2.1).
openssl is already the newest version (1.1.1f-1ubuntu2.1).
0 upgraded, 0 newly installed, 0 to remove and 118 not upgraded.
```

⑥ 安装 Nginx，在当前目录下执行 make 命令，然后在执行 make install 命令；

⑦ 启动 Nginx，进入 Nginx 的安装目录，也就是刚才我们配置的/usr/local/nginx 下的 sbin 目录下执行./nginx；



```
root@iZh846luthz90jZ:/# cd usr
root@iZh846luthz90jZ:/usr# ls
bin  games  include  lib  lib32  lib64  libx32  local  nginx  sbin  share  src
root@iZh846luthz90jZ:/usr# cd nginx
root@iZh846luthz90jZ:/usr/nginx# ls
nginx-1.18.0  nginx-1.18.0.tar.gz
root@iZh846luthz90jZ:/usr/nginx# cd nginx-1.18.0/
root@iZh846luthz90jZ:/usr/nginx/nginx-1.18.0# ls
auto  CHANGES  CHANGES.ru  conf  configure  contrib  html  LICENSE  Makefile  man  objs  README  src
```

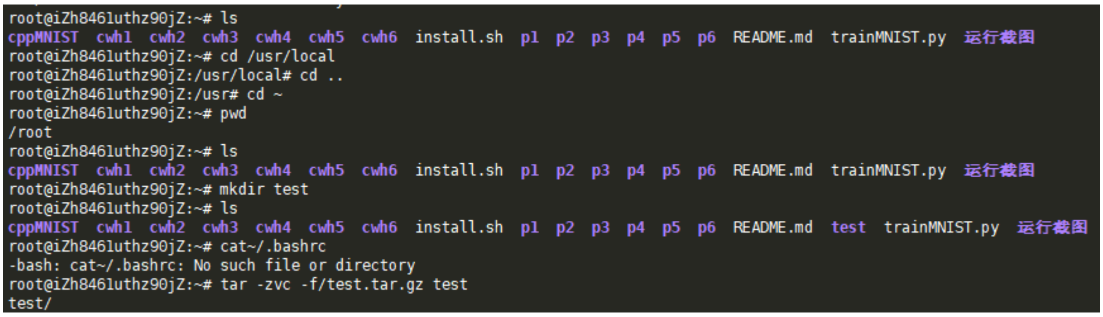


```
root@iZh846luthz90jZ:/usr# cd local
root@iZh846luthz90jZ:/usr/local# ls
aegis  bin  etc  games  include  lib  man  nginx  sbin  share  src
root@iZh846luthz90jZ:/usr/local# cd nginx
root@iZh846luthz90jZ:/usr/local/nginx# ls
conf  html  sbin
root@iZh846luthz90jZ:/usr/local/nginx# cd sbin
root@iZh846luthz90jZ:/usr/local/nginx/sbin# ./nginx
```

⑧ 下载搜狗数据集，根据资料中的 vim 命令进行操作。

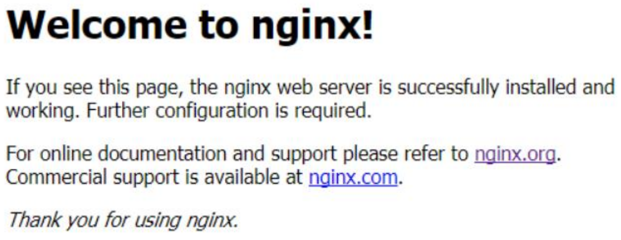
（3） 运行结果

① 命令行指令操作



```
root@iZh846luthz90jZ:~# ls
cppMNIST  cwh1  cwh2  cwh3  cwh4  cwh5  cwh6  install.sh  p1  p2  p3  p4  p5  p6  README.md  trainMNIST.py  运行截图
root@iZh846luthz90jZ:~# cd /usr/local
root@iZh846luthz90jZ:/usr/local# cd ..
root@iZh846luthz90jZ:/usr# cd ~
root@iZh846luthz90jZ:~# pwd
/root
root@iZh846luthz90jZ:~# ls
cppMNIST  cwh1  cwh2  cwh3  cwh4  cwh5  cwh6  install.sh  p1  p2  p3  p4  p5  p6  README.md  trainMNIST.py  运行截图
root@iZh846luthz90jZ:~# mkdir test
root@iZh846luthz90jZ:~# ls
cppMNIST  cwh1  cwh2  cwh3  cwh4  cwh5  cwh6  install.sh  p1  p2  p3  p4  p5  p6  README.md  test  trainMNIST.py  运行截图
root@iZh846luthz90jZ:~# cat~/.bashrc
-bash: cat~/.bashrc: No such file or directory
root@iZh846luthz90jZ:~# tar -zvc -f/test.tar.gz test
test/
```

② Nginx 运行效果



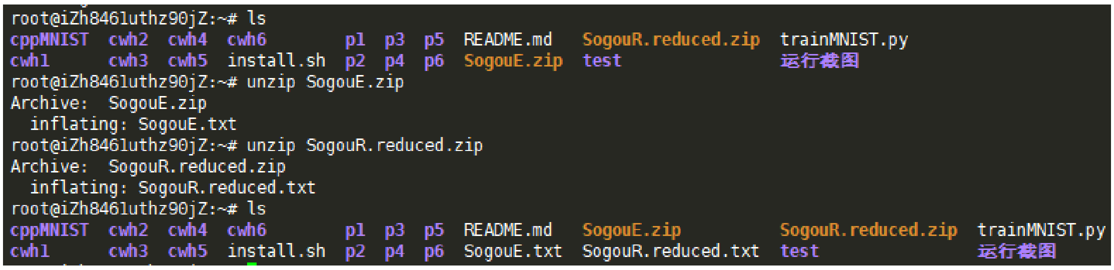
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

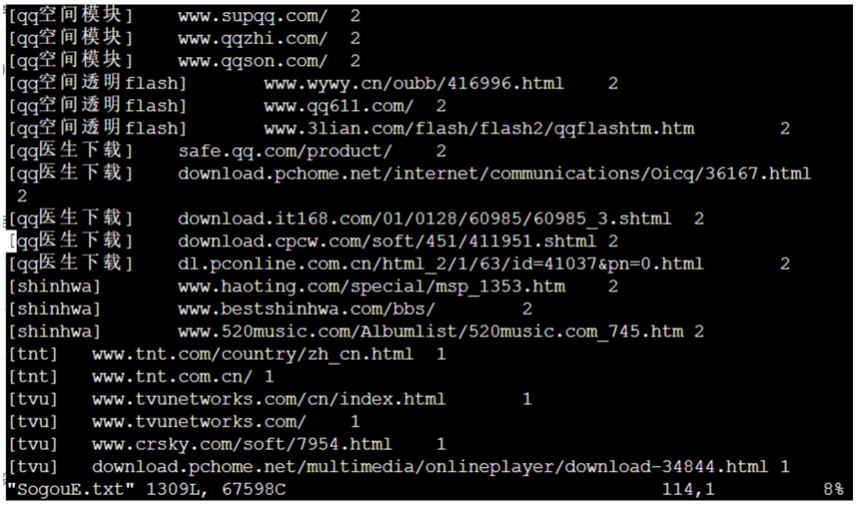
Thank you for using nginx.

③ 对搜狗数据集的操作



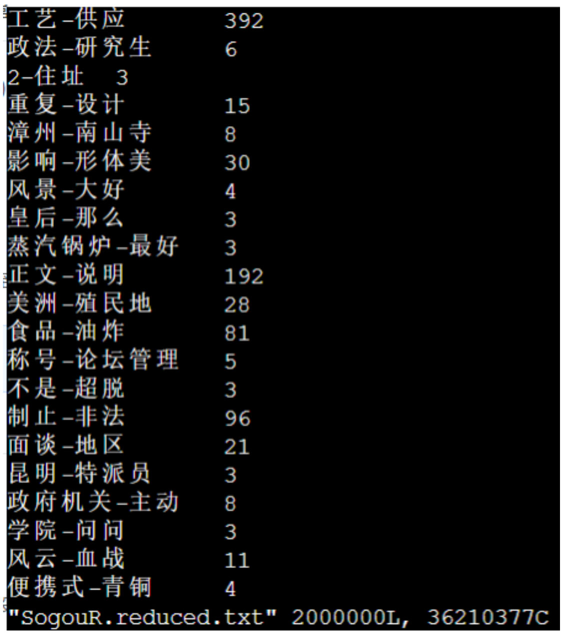
```
root@iZh846luthz90jZ:~# ls
cppMNIST  cwh2  cwh4  cwh6        p1  p3  p5  README.md  SogouR.reduced.zip  trainMNIST.py
cwh1      cwh3  cwh5  install.sh  p2  p4  p6  SogouE.zip  test               运行截图
root@iZh846luthz90jZ:~# unzip SogouE.zip
Archive:  SogouE.zip
  inflating: SogouE.txt
root@iZh846luthz90jZ:~# unzip SogouR.reduced.zip
Archive:  SogouR.reduced.zip
  inflating: SogouR.reduced.txt
root@iZh846luthz90jZ:~# ls
cppMNIST  cwh2  cwh4  cwh6        p1  p3  p5  README.md   SogouE.zip          SogouR.reduced.zip  trainMNIST.py
cwh1      cwh3  cwh5  install.sh  p2  p4  p6  SogouE.txt  SogouR.reduced.txt  test                运行截图
```

解压压缩包



```
[qq空间模块]    www.supqq.com/  2
[qq空间模块]    www.qqzhi.com/  2
[qq空间模块]    www.qqson.com/  2
[qq空间透明flash]       www.wywy.cn/oubb/416996.html    2
[qq空间透明flash]       www.qq611.com/  2
[qq空间透明flash]       www.3lian.com/flash/flash2/qqflashtm.htm        2
[qq医生下载]    safe.qq.com/product/    2
[qq医生下载]    download.pchome.net/internet/communications/Oicq/36167.html
 2
[qq医生下载]    download.it168.com/01/0128/60985/60985_3.shtml  2
[qq医生下载]    download.cpcw.com/soft/451/411951.shtml 2
[qq医生下载]    dl.pconline.com.cn/html_2/1/63/id=41037&pn=0.html       2
[shinhwa]       www.haoting.com/special/msp_1353.htm    2
[shinhwa]       www.bestshinhwa.com/bbs/        2
[shinhwa]       www.520music.com/Albumlist/520music.com_745.htm 2
[tnt]   www.tnt.com/country/zh_cn.html  1
[tnt]   www.tnt.com.cn/ 1
[tvu]   www.tvunetworks.com/cn/index.html       1
[tvu]   www.tvunetworks.com/    1
[tvu]   www.crsky.com/soft/7954.html    1
[tvu]   download.pchome.net/multimedia/onlineplayer/download-34844.html 1
"SogouE.txt" 1309L, 67598C                                    114,1         8%
```

编辑 SogouE.txt



```
工艺-供应        392
政法-研究生      6
2-住址   3
重复-设计        15
漳州-南山寺      8
影响-形体美      30
风景-大好        4
皇后-那么        3
蒸汽锅炉-最好    3
正文-说明        192
美洲-殖民地      28
食品-油炸        81
称号-论坛管理    5
不是-超脱        3
制止-非法        96
面谈-地区        21
昆明-特派员      3
政府机关-主动    8
学院-问问        3
风云-血战        11
便携式-青铜      4
"SogouR.reduced.txt" 2000000L, 36210377C
```

编辑 SogouR.reduced.txt

3.总结

第二次实验主要巩固了 Linux 命令行的操作，我更多的关注了 Nginx 的工作原理，因为在了解到可以通过 Nginx 建立自己的博客后我对他十分感兴趣。

最开始，我是看的理论，我认为学习一门技术首先要了解它到底是什么，到底能做什么，能给我们带来什么它的好处，带着这种疑问进行学习是简单且效率最高的，我从网上查阅了一些资料，使我了解了反向代理和负载均衡。

反向代理就是以代理服务器来接受 internet 上的连接请求，然后将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给 internet 上请求连接的客户端，此时代理服务器对外就表现为一个服务器。

负载均衡其实就是将流量分发到多个服务器上执行，减轻每台服务器的压力，多台服务器共同完成工作任务，从而提高了数据的吞吐量，从而扩展了网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性。

使用 Nginx 我们可以做到动静分离，将万年不动的静态资源放到 Nginx 中，而动态资源运行在 TomCat 服务器中，当访问静态资源时，直接请求 Nginx 就可以了，不在需要去请求 TomCat 这样服务器的压力又小了

当我了解完这些的时候，心情很激动，因为现在目前系统是都是多个用户访问同一台服务器，可想而知服务器的压力有多大，但是如果使用 Nginx 就可以利用 Nginx 做一个负载均衡，搭建一个集群，这样将数据流量分发了下去，处理速度不光快了，服务器压力也小了，

一点点的优化系统会越来越流畅，就像我们学习，一点点积累，一点点沉淀就会发现自己视野越来越宽，学习越来越轻松

三、Shell 程序设计

1.设计内容

（1）设计如下一个菜单驱动程序。

Use one of the following options:

P: To display current directory

S: To display the name of running file

D: To display today’s date and present time

L: To see the listing of files in your present working directory

W: To see who is logged in

Q: To quit this program

Enter your option and hit:

菜单程序将根据用户输入的选择项给出相应信息。要求对用户的输入忽略大小写，对于无效选项的输入给出相应提示。要求使用 case 语句实现以上功能，输入相应的字母后应该执行相应的命令完成每项功能，如输入 P 或 p，就执行 pwd 命令。

（2）编写一段 bash Shell 程序，根据从键盘输入的学生成绩，显示相应的成绩等级，其中 60 分以下为“Failed!”，60~70 分为“Passed!”，70~80 分为“Medium!”，80~90 分为“Good!”，90~100 分为“Excellent!”。如果输入超过 100 或负数的分数，则显示错误分数提示。

（3）编写一个 Shell 过程完成如下功能（必须在脚本中使用函数）。

① 程序接收 3 个参数：$1、$2 和$3，合并两个文件$1、$2 为$3，并显示，三个文件均为文本文件。

② 如果文件$3 不存在，那么先报告缺少$3，然后将合并后的内容输出到 mydoc.txt。如果有$3，就合并到$3。

④ 如果文件$2 和文件$3 都不存在，那么先报告缺少$2、$3，只显示$1 的内容。

⑤ 如果文件$1 不存在，则提示缺少$1，要求重新运行程序。

（4）编写一个脚本，显示当天日期，查找给定的某用户是否在系统中工作。如果在系统中，就发一个问候给他。

（5）给 shell 增加一个新的命令 mycommand（自己起一个名字）：

编程实现某 shell 命令的功能，如 ls 命令，显示当前目录下的目录及文件信息。同学们也可以编写不同的命令功能，如 cp，rm 等。必须编程实现，不能直接调用该命令的功能。

上述步骤实现后，为自己的命令增加至少两个选项（option），如 ls 命令可增加一个-l 的选项，查看文件及目录的详细信息。

将自定义的命令设置环境变量和权限，以 root 登录时在任何目录中均可正确执行该命令，同时退出登录再次登录时也可以在任何目录使用该命令。

2.具体设计

（1） 设计思路

① 是用类似 switch 的语句进行对用户的指令进行分类

② 使用 shell 语句对用户输入的成绩进行区间的判断，在判断结束后对结果对应所属区间的反应

③ 根据函数中三个参数存在与否对不同的情况作出不同的判断

④ 编写一个脚本，显示当天日期，查找给定的某用户是否在系统中工作。如果在系统中，就发一个问候给他。

⑤ 编写程序，实现某种功能，在实现功能后对其添加选项，使其可以通过不同的指令做出不同的操作

⑥ 添加命令的主要参数与功能为：

cwh -a：显示当前路径下的所有文件

cwh -q：显示时间、大小、文件名、路径

cwh -u "name"：显示所有文件作者是"name"的文件

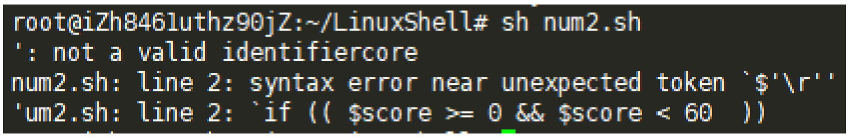
cwh -R：递归显示当前路径下的的子目录中的文件和文件夹

cwh -1：每行显示一个文件

cwh -s：显示每个文件或文件夹下有多少文件

（2） 操作步骤

① 为了解决将代码在 Windows 操作系统编辑后在 Linux 操作系统重新运行时句尾会出现换行符以及空行会出现空行符的问题，需要首先使用命令：sed -i 's/\r//' xxx.sh；



```
root@iZh8461uthz90jZ:~/LinuxShell# sh num2.sh
': not a valid identifiercore
num2.sh: line 2: syntax error near unexpected token `$'\r''
'um2.sh: line 2: `if (( $score >= 0 && $score < 60  ))
```

② 使用 sh 指令运行编写好的 shell 脚本。

③ 如果程序是 C 语言，则首先编译源码，命令如下：

```
gcc name.c -o target
```

④ 将编译好的可执行程序添加到环境变量，使用如下命令：

```
cp a /usr/local/bin
```

（3） 具体代码

① 第一题

```
echo 'Use one of the following options:'
echo '    P: To display current directory'
echo '    S: To display the name of running file'
echo '    D: To display today’s date and present time'
echo '    L: To see the listing of files in your present working directory'
echo '    W: To see who is logged in'
echo '    Q: To quit this program'
read -n 1 -p 'Enter your option and hit: '    option
echo "Your option is $option"
case ${option^^} in
        P)
                pwd
                ;;
        S)
                echo $(basename $0)
                ;;
        D)
                date
                ;;
        L)
```

```
                ls -al
                ;;
        W)
                whoami
                ;;
        Q)
                exit
                ;;
        *)
                echo 'Please enter as required!'
                ;;
Esac
```

② 第二题

```
read -p "Input your score: " score
if (( $score >= 0 && $score < 60    ))
then
        echo 'Failed!'
elif (( $score >= 60 && $score < 70 ))
then
        echo 'Passed!'
elif (( $score >= 70 && $score < 80 ))
then
        echo 'Medium!'
elif (( $score >= 80 && $score < 90 ))
then
        echo 'Good!'
elif (( $score >= 90 && $score <= 100 ))
then
        echo 'Excellent!'
```

```
else
        echo 'Input error!'
fi
```

③ 第三题

```
#!/bin/sh
function f1(){
        echo "至少输入 2 个参数"
        cat $1
}
function f2(){
        echo "缺少第 3 个参数，输出到 mydoc.txt"
        cat $1 $2 > mydoc.txt
        cat mydoc.txt
}

function Usage(){
        echo "Usage:$0 <arg1> <arg2> [arg3]"
}
if [ $# -eq 0 ]; then
        Usage
elif [ $# -eq 1 ]; then
        f1 $1
elif [ $# -eq 2 ]; then
        f2 $1 $2
elif [ $# -eq 3 ]; then
        cat $1 $2 > $3
        cat $3
fi
exit 0
```

④ 第四题

```
date
echo   “Input   a   user   name   for   searching”
read   name
if   who|grep   “$name”
then
   echo   “Lucky, there   is   $name in   my   system   now.”
   echo   “Hi,how are you!” > /var/spool/mail/$name
else   echo   “Unlucky, there   is not   $name in   my   system.”
fi
```

⑤ 第五题

```
struct ConBit {
      unsigned int cb, sort; //control block
} CB;

/*
#define S_ISFIFO( m ) ( ( (m) & (S_IFMT) ) == ( 0010000) )
#define S_ISDIR( m )    ( ( (m) & (S_IFMT) ) == (0040000))
#define S_ISCHR(m) ( ( (m) & (S_IFMT) ) ==( 0020000))
#define S_ISBLK(m) ( ( (m) & (S_IFMT) ) ==(0060000))
#define S_ISREG(m) ( ( (m) & (S_IFMT) ) ==(0100000))
  */

struct ls {
      char fname[256];
      struct stat info;
      struct ls** next;
      struct ls** back;
};
```

```
char *b_opt_arg; //argument of options
int cnt;
struct ls **buffer, **head, **tail;
//***************Compare Block**************************************************

int cmp_nlink(const void * a, const void * b) {
    return ((struct ls*) a)->info.st_nlink - ((struct ls*) b)->info.st_nlink;
}

int cmp_st_size(const void * a, const void * b) {
    return ((struct ls*) a)->info.st_size - ((struct ls*) b)->info.st_size;
}

int cmp_mtim(const void * a, const void * b) {
    return -(int) (((struct ls*) a)->info.st_mtim.tv_sec - ((struct ls*) b)->info.st_mtim.tv_sec);
}

int cmp_fname(const void * a, const void * b) {
    int t = strcmp(((struct ls*) a)->fname, ((struct ls*) b)->fname);
    return t > 0 ? 1 : -1;
}

void InsertSort(struct ls** pData, int ( *compare)(const void* a, const void* b)) {
    if (pData == NULL) return;
    struct ls** p;
    struct ls** q;
    for (p = (*pData)->next; p != NULL; p = (*p)->next) {
        for (q = (*p)->back; q != NULL && (compare(*p, *q)) < 0; q = (*q)->back);
        (*((*p)->back))->next = (*p)->next;
        if ((*p)->next)
            (*((*p)->next))->back = (*p)->back;

        if (q) {
            (*p)->back = q;
            (*p)->next = (*q)->next;
            if ((*p)->next)
                (*((*q)->next))->back = p;
            (*q)->next = p;
        } else {
            (*p)->back = NULL;
            (*p)->next = head;
            (*head)->back = p;
            head = p;
        }
    }
}

/*
void Qsort ( void * begin_t ,void *end_t   ,   int ( *compare)(const void* a, const void* b ) )
{
    struct ls **low=(struct ls**)begin_t,**up = (struct ls**)end_t , **p,**q,**t;
    if ( low!=up )
    {
        p=low,q=up;
        t=low;
        while ( p!=q )
        {
```

```
                        while ( p!=q && compare(*q,*t) )
                                q=(*q)->back;
                        if ( p!=q ) {
                                p=q;
                                p=(*p)->next;
                        }
                }
        }
}
  */

char numbuf[16];

char* uid_to_name(uid_t uid) {
        if (getpwuid(uid)) {
                char* pw_ptr = getpwuid(uid)->pw_name;
                if (pw_ptr)
                        return pw_ptr;
        }
        fprintf(numbuf, "%d", uid);
        return numbuf;
}

char* gid_to_name(gid_t gid) {
        char *gr_ptr = getgrgid(gid)->gr_name;
        if (gr_ptr)
                return gr_ptr;
        char numstr[10];
        fprintf(numstr, "%d", gid);
        return numstr;
```

```
}


void mode_to_letters(int mode, char *str) {
    strcpy(str, "----------");


    if (S_ISDIR(mode)) str[0] = 'd';
    if (S_ISCHR(mode)) str[0] = 'c';
    if (S_ISBLK(mode)) str[0] = 'b';


    if (mode & S_IRUSR) str[1] = 'r';
    if (mode & S_IWUSR) str[2] = 'w';
    if (mode & S_IXUSR) str[3] = 'x';


    if (mode & S_IRGRP) str[4] = 'r';
    if (mode & S_IWGRP) str[5] = 'w';
    if (mode & S_IXGRP) str[6] = 'x';


    if (mode & S_IROTH) str[7] = 'r';
    if (mode & S_IWOTH) str[8] = 'w';
    if (mode & S_IXOTH) str[9] = 'x';
}


void store_file_info(char* fname, struct stat* buf) {


    if (fname[0] == '.' && !(CB.cb & ISa)) return;
    struct ls** p = (struct ls**) (malloc(sizeof (struct ls*)));
    *p = (struct ls*) malloc(sizeof (struct ls));
    (*p)->next = NULL;
    (*p)->back = buffer;
    (*p)->info = (*buf);
    strcpy((*p)->fname, fname);
    if (buffer == NULL) {
        head = buffer = p;
    } else {
        (*buffer)->next = p;
        buffer = p;
    }
    tail = buffer;
    cnt++;

}

void dostat(char* fname) {
    struct stat info;
    char *namep;
    if (stat(fname, &info) == -1) {
        perror(fname);
        return;
    } else {
        namep = fname;
        fname = strrchr(namep, '/'); //cut prefix
        store_file_info(fname == 0 ? namep : fname + 1, &info);
        return;
    }
}

void sort() {
    if (!(CB.cb & ISS)) return;
    switch (CB.sort) {
        case BYMTIME:
            InsertSort(buffer, cmp_mtim);
            break;
        case BYFNAME:
            InsertSort(buffer, cmp_fname);
            break;
        case BYSTSIZE:
            InsertSort(buffer, cmp_st_size);
            break;
        case BYNLINK:
            InsertSort(buffer, cmp_nlink);
            break;
        default:
            break;
    }
}

void show_file_info() {
    //printf("Block Number=%d,        Number of Directions(files)=%d\n\n", block, cnt);
    void mode_to_letters(int, char*);
    char* uid_to_name(uid_t);
    char* gid_to_name(gid_t);
    int i, j, block = 0;
    struct ls** p;
    char strmode[11], strusr[11], strgrp[11];
    // show -l list
    if (CB.cb & ISl) {
        for (j = 0, p = head; p; p = (*p)->next) {

            strcpy(strusr, uid_to_name((*p)->info.st_uid));
            strcpy(strgrp, gid_to_name((*p)->info.st_gid));
            if (CB.cb & ISu && strcmp(strusr, b_opt_arg)) continue;
            if (CB.cb & ISs)
                printf("%04d", (*p)->info.st_blocks / 2);
            mode_to_letters((*p)->info.st_mode, strmode);
            printf("%8s", strmode);
            printf("%4d", (int) (*p)->info.st_nlink);
            printf("    %-8s", strusr);
            printf("%-8s", strgrp);
            printf("%8ld", (*p)->info.st_size);
            printf("    %.12s", 4 + ctime(&(*p)->info.st_mtim));
            if (S_ISDIR((*p)->info.st_mode))
                printf(FORMAT_LDIR, (*p)->fname);
            else
                if (S_ISREG((*p)->info.st_mode))
                printf(FORMAT_LREG, (*p)->fname);
            else
                printf("    %-8s\n", (*p)->fname);
            block += (*p)->info.st_blocks;
            j++;

        }
        printf("\nTotal files(directors) = %d, and Total Block = %d\n", j, block / 2);
        return;
    }
    // show simple list
    if (CB.cb & ISC) {

    } else {
        for (j = 0, p = head; p; p = (*p)->next) {
            strcpy(strusr, uid_to_name((*p)->info.st_uid));


            if (CB.cb & ISu && strcmp(strusr, b_opt_arg)) continue;
            if (j && !(CB.cb & IS1))
                printf("        ");
            if (CB.cb & ISs)
                printf("%d ", (*p)->info.st_blocks / 2);
            if (S_ISDIR((*p)->info.st_mode))
                printf(FORMAT_SDIR, (*p)->fname);
            else
                if (S_ISREG((*p)->info.st_mode))
                printf(FORMAT_SREG, (*p)->fname);
            else
                printf("%s", (*p)->fname);
            if (CB.cb & IS1)
                puts("");
            block += (*p)->info.st_blocks / 2;
            j++;
        }
        if (CB.cb & ISs)
            printf("\nTotal Blocks = %d", block);
        printf("\nTotal files(directors) = %d.\n", j);
    }
}

void do_ls(char* dirname) {
    DIR *dir_ptr;
    struct dirent * direntp;
    char *fpath;
    int i;
```

```
struct ls** p;
struct ls** t;
struct ls* q;
if ((dir_ptr = opendir(dirname)) == NULL)
        fprintf(stderr, "myls:cannot open %s\n", dirname);
else {
        cnt = 0;
        fpath = (char*) malloc(strlen(dirname) + 1 + MAXNAMESIZE + 1);
        while ((direntp = readdir(dir_ptr)) != NULL) {
                strcpy(fpath, dirname);
                strcat(fpath, "/");
                strcat(fpath, direntp->d_name);
                dostat(fpath);
        }
        buffer = head;
        free(fpath);
        closedir(dir_ptr);
        sort();
        show_file_info();
        /////////////release memory//////////
        for (p = head; p;) {
                q = (*p);
                t = q->next;
                free(q);
                free(p);
                p = t;
        }
        head = buffer = 0;
        ////////////////////////////////////
}
```

```
}


void rec_show(char* dirname) {
    DIR *dir_ptr;
    struct stat info;
    struct dirent * direntp;
    char *fpath;
    int i;
    printf("\033[1;32m%s:\n\033[0m", dirname);
    do_ls(dirname);

    printf("\n\n");
    if ((dir_ptr = opendir(dirname)) == NULL) {
        fprintf(stderr, "myls:cannot open %s\n", dirname);
        return;
    }
    fpath = (char*) malloc(strlen(dirname) + 1 + MAXNAMESIZE + 1);
    while ((direntp = readdir(dir_ptr)) != NULL) {
        if (!(CB.cb & ISa) && direntp->d_name[0] == '.') continue;
        if (strcmp(direntp->d_name, ".") && strcmp(direntp->d_name, "..")) {
            strcpy(fpath, dirname);
            strcat(fpath, "/");
            strcat(fpath, direntp->d_name);
            if (stat(fpath, &info) == -1) {
                perror(fpath);
                break;
            } else {
                if (S_ISDIR(info.st_mode)) {
                    rec_show(fpath);
                }
```

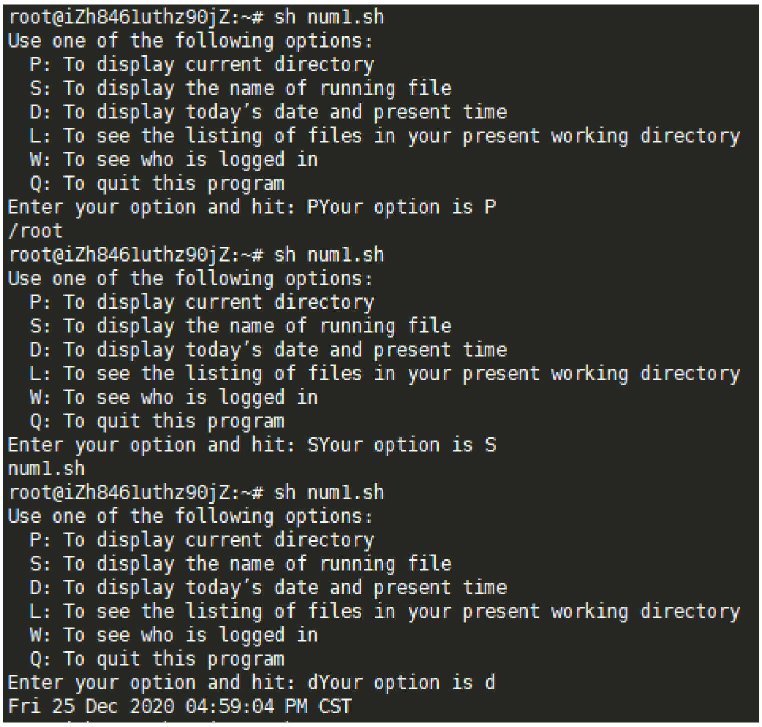
```
                        }


                }

        }

        free(fpath);

        closedir(dir_ptr);

}


int main(int argc, char **argv) {


        int oc;

        char* dir = NULL;

        head = buffer = NULL;

        if (argc > 1 && argv[1][0] == '/') {

                dir = argv[1];

                argc--;

                argv++;

        }

        CB.cb = CB.sort = 0;

        while ((oc = getopt(argc, argv, "u:RlaS:1s")) != -1) {

                switch (oc) {

                        case 'S':

                                CB.cb |= ISS;

                                b_opt_arg = optarg;

#ifdef DEBUG

                                printf("the option is -q\nthe sort way is %s\n", optarg);

#endif

                                if (strcmp("time", optarg) == 0)

                                        CB.sort = BYMTIME;

                                else
                                        if (strcmp("size", optarg) == 0)
                                        CB.sort = BYSTSIZE;
                                else
                                        if (strcmp("link", optarg) == 0)
                                        CB.sort = BYNLINK;
                                else
                                        if (strcmp("name", optarg) == 0)
                                        CB.sort = BYFNAME;
                                else {
                                        printf("arguments error!\n");
                                        return 1;
                                }
#ifdef DEBUG
                                printf("%d\n", CB.cb);
#endif
                                break;
                        case 'a':
#ifdef DEBUG
                                printf("the option is -a\n");
#endif

                                CB.cb |= ISa;
                                break;
                        case 'l':
#ifdef DEBUG
                                printf("the option is -l\n");
#endif

                                CB.cb |= ISl;
                                break;
```

```
			case 'u':
				b_opt_arg = optarg;
#ifdef DEBUG
				printf("the option is -u\nfilename is %s\n", optarg);
#endif

				CB.cb |= ISu;
				break;
			case 'R':
#ifdef DEBUG
				printf("the option is -R\n");
#endif

				CB.cb |= ISR;
				break;

			case 'C':
#ifdef DEBUG
				printf("the option is -c\n");
#endif
				CB.cb |= ISC;
				break;
			case '1':
				CB.cb |= IS1;
				break;
			case 's':
				CB.cb |= ISs;
				break;
			case '?':
				printf("arguments error!\n");
```

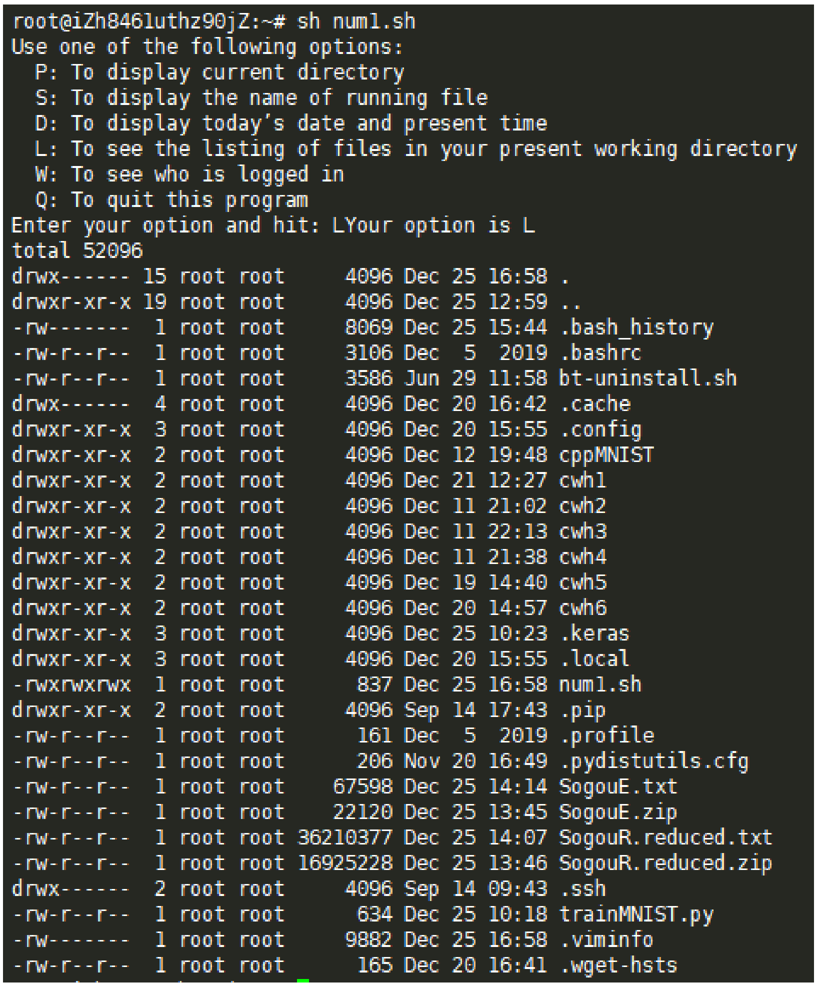
```
                break;
        }
    }
    if (CB.cb & ISR) {
        if (dir)
            rec_show(dir);
        else
            rec_show(".");
    } else {
        if (dir)
            do_ls(dir);
        else
            do_ls(".");
    }
    return 0;
}
```

（4） 运行结果

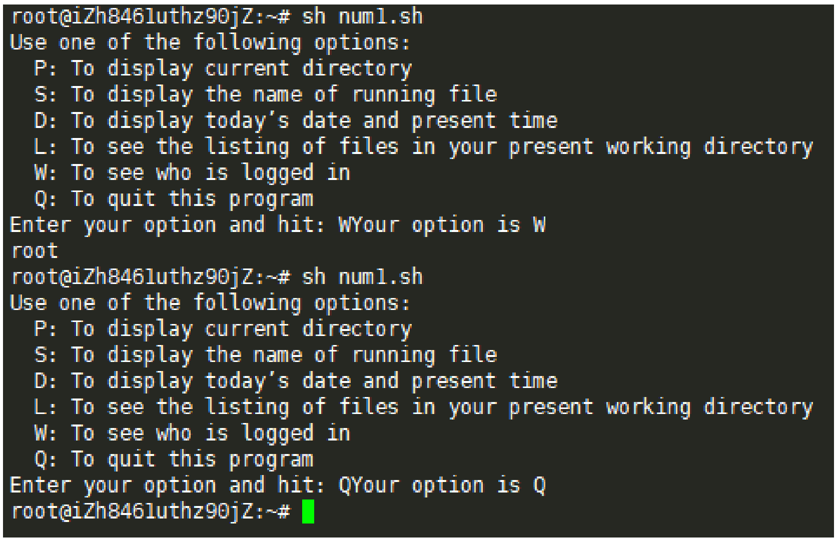
① 第一题



```
root@iZh846luthz90jZ:~# sh num1.sh
Use one of the following options:
  P: To display current directory
  S: To display the name of running file
  D: To display today’s date and present time
  L: To see the listing of files in your present working directory
  W: To see who is logged in
  Q: To quit this program
Enter your option and hit: PYour option is P
/root
root@iZh846luthz90jZ:~# sh num1.sh
Use one of the following options:
  P: To display current directory
  S: To display the name of running file
  D: To display today’s date and present time
  L: To see the listing of files in your present working directory
  W: To see who is logged in
  Q: To quit this program
Enter your option and hit: SYour option is S
num1.sh
root@iZh846luthz90jZ:~# sh num1.sh
Use one of the following options:
  P: To display current directory
  S: To display the name of running file
  D: To display today’s date and present time
  L: To see the listing of files in your present working directory
  W: To see who is logged in
  Q: To quit this program
Enter your option and hit: dYour option is d
Fri 25 Dec 2020 04:59:04 PM CST
```

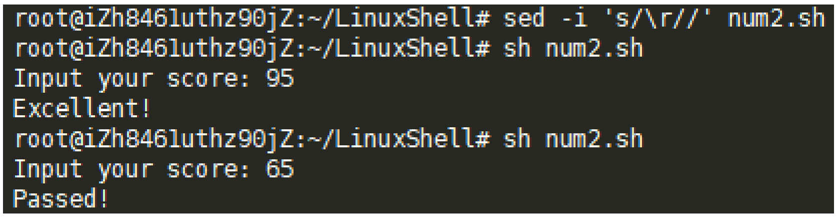


```
root@iZh846luthz90jZ:~# sh num1.sh
Use one of the following options:
  P: To display current directory
  S: To display the name of running file
  D: To display today's date and present time
  L: To see the listing of files in your present working directory
  W: To see who is logged in
  Q: To quit this program
Enter your option and hit: LYour option is L
total 52096
drwx------ 15 root root     4096 Dec 25 16:58 .
drwxr-xr-x 19 root root     4096 Dec 25 12:59 ..
-rw-------  1 root root     8069 Dec 25 15:44 .bash_history
-rw-r--r--  1 root root     3106 Dec  5  2019 .bashrc
-rw-r--r--  1 root root     3586 Jun 29 11:58 bt-uninstall.sh
drwx------  4 root root     4096 Dec 20 16:42 .cache
drwxr-xr-x  3 root root     4096 Dec 20 15:55 .config
drwxr-xr-x  2 root root     4096 Dec 12 19:48 cppMNIST
drwxr-xr-x  2 root root     4096 Dec 21 12:27 cwh1
drwxr-xr-x  2 root root     4096 Dec 11 21:02 cwh2
drwxr-xr-x  2 root root     4096 Dec 11 22:13 cwh3
drwxr-xr-x  2 root root     4096 Dec 11 21:38 cwh4
drwxr-xr-x  2 root root     4096 Dec 19 14:40 cwh5
drwxr-xr-x  2 root root     4096 Dec 20 14:57 cwh6
drwxr-xr-x  3 root root     4096 Dec 25 10:23 .keras
drwxr-xr-x  3 root root     4096 Dec 20 15:55 .local
-rwxrwxrwx  1 root root      837 Dec 25 16:58 num1.sh
drwxr-xr-x  2 root root     4096 Sep 14 17:43 .pip
-rw-r--r--  1 root root      161 Dec  5  2019 .profile
-rw-r--r--  1 root root      206 Nov 20 16:49 .pydistutils.cfg
-rw-r--r--  1 root root    67598 Dec 25 14:14 SogouE.txt
-rw-r--r--  1 root root    22120 Dec 25 13:45 SogouE.zip
-rw-r--r--  1 root root 36210377 Dec 25 14:07 SogouR.reduced.txt
-rw-r--r--  1 root root 16925228 Dec 25 13:46 SogouR.reduced.zip
drwx------  2 root root     4096 Sep 14 09:43 .ssh
-rw-r--r--  1 root root      634 Dec 25 10:18 trainMNIST.py
-rw-------  1 root root     9882 Dec 25 16:58 .viminfo
-rw-r--r--  1 root root      165 Dec 20 16:41 .wget-hsts
```



```
root@iZh846luthz90jZ:~# sh num1.sh
Use one of the following options:
  P: To display current directory
  S: To display the name of running file
  D: To display today's date and present time
  L: To see the listing of files in your present working directory
  W: To see who is logged in
  Q: To quit this program
Enter your option and hit: WYour option is W
root
root@iZh846luthz90jZ:~# sh num1.sh
Use one of the following options:
  P: To display current directory
  S: To display the name of running file
  D: To display today's date and present time
  L: To see the listing of files in your present working directory
  W: To see who is logged in
  Q: To quit this program
Enter your option and hit: QYour option is Q
root@iZh846luthz90jZ:~# 
```

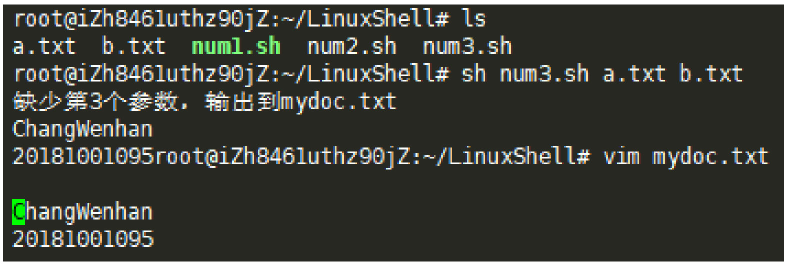
② 第二题



```
root@iZh846luthz90jZ:~/LinuxShell# sed -i 's/\r//' num2.sh
root@iZh846luthz90jZ:~/LinuxShell# sh num2.sh
Input your score: 95
Excellent!
root@iZh846luthz90jZ:~/LinuxShell# sh num2.sh
Input your score: 65
Passed!
```

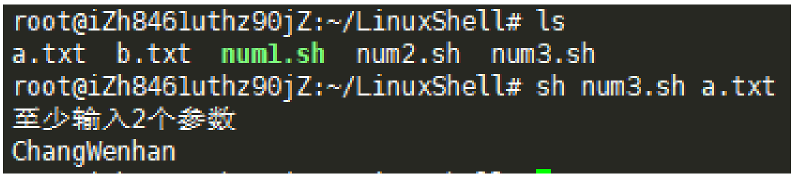
③ 第三题

如果文件$3 不存在，那么先报告缺少$3，然后将合并后的内容输出到 mydoc.txt。如果有$3，就合并到$3。



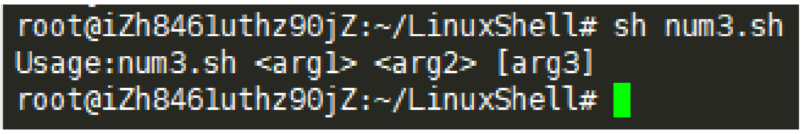
```
root@iZh846luthz90jZ:~/LinuxShell# ls
a.txt  b.txt  num1.sh  num2.sh  num3.sh
root@iZh846luthz90jZ:~/LinuxShell# sh num3.sh a.txt b.txt
缺少第3个参数，输出到mydoc.txt
ChangWenhan
20181001095root@iZh846luthz90jZ:~/LinuxShell# vim mydoc.txt

ChangWenhan
20181001095
```

如果文件$2 和文件$3 都不存在，那么先报告缺少$2、$3，只显示$1 的内容。



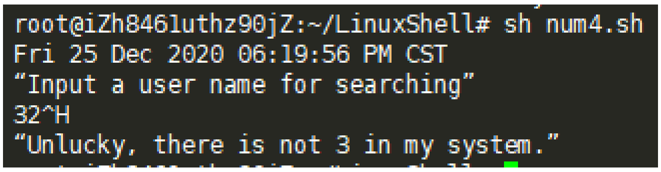
```
root@iZh846luthz90jZ:~/LinuxShell# ls
a.txt  b.txt  num1.sh  num2.sh  num3.sh
root@iZh846luthz90jZ:~/LinuxShell# sh num3.sh a.txt
至少输入2个参数
ChangWenhan
```

如果文件$1 不存在，则提示缺少$1，要求重新运行程序。



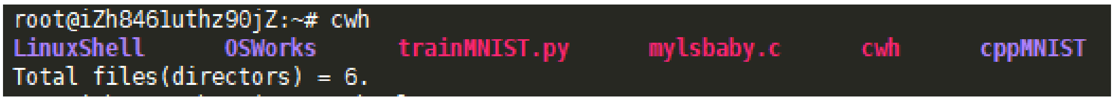
```
root@iZh846luthz90jZ:~/LinuxShell# sh num3.sh
Usage:num3.sh <arg1> <arg2> [arg3]
root@iZh846luthz90jZ:~/LinuxShell# 
```

④ 第四题

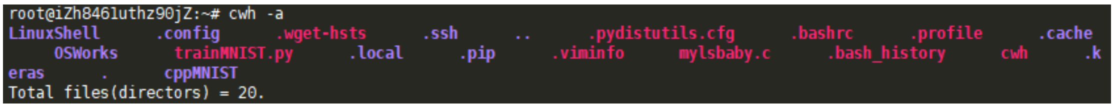


```
root@iZh846luthz90jZ:~/LinuxShell# sh num4.sh
Fri 25 Dec 2020 06:19:56 PM CST
“Input a user name for searching”
32^H
“Unlucky, there is not 3 in my system.”
```

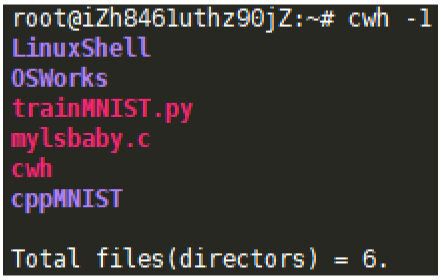
⑤ 第五题



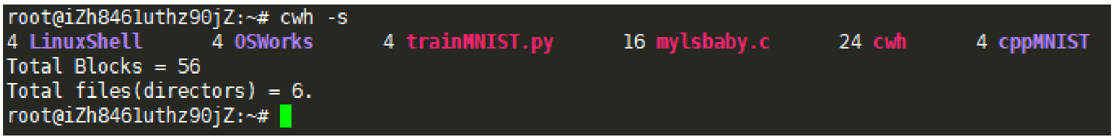
```
root@iZh846luthz90jZ:~# cwh
LinuxShell       OSWorks       trainMNIST.py       mylsbaby.c       cwh       cppMNIST
Total files(directors) = 6.
```



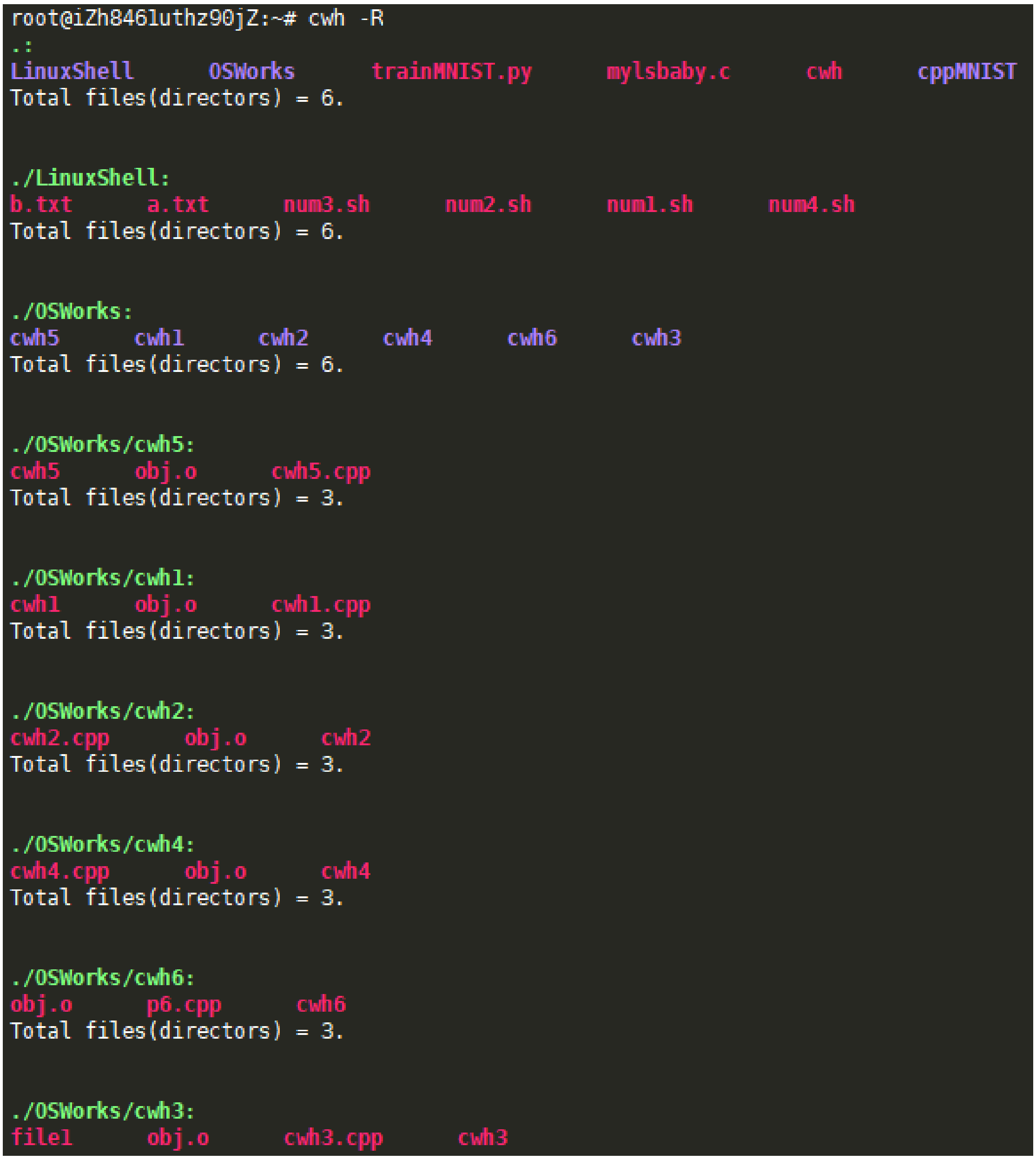
```
root@iZh846luthz90jZ:~# cwh -a
LinuxShell      .config      .wget-hsts      .ssh      ..      .pydistutils.cfg      .bashrc      .profile      .cache
     OSWorks      trainMNIST.py      .local      .pip      .viminfo      mylsbaby.c      .bash_history      cwh      .k
eras      .      cppMNIST
Total files(directors) = 20.
```



```
root@iZh846luthz90jZ:~# cwh -l
LinuxShell
OSWorks
trainMNIST.py
mylsbaby.c
cwh
cppMNIST

Total files(directors) = 6.
```



```
root@iZh846luthz90jZ:~# cwh -s
4 LinuxShell      4 OSWorks      4 trainMNIST.py      16 mylsbaby.c      24 cwh      4 cppMNIST
Total Blocks = 56
Total files(directors) = 6.
root@iZh846luthz90jZ:~#
```



```
root@iZh846luthz90jZ:~# cwh -R
.:
LinuxShell      OSWorks      trainMNIST.py      mylsbaby.c      cwh      cppMNIST
Total files(directors) = 6.


./LinuxShell:
b.txt      a.txt      num3.sh      num2.sh      num1.sh      num4.sh
Total files(directors) = 6.


./OSWorks:
cwh5      cwh1      cwh2      cwh4      cwh6      cwh3
Total files(directors) = 6.


./OSWorks/cwh5:
cwh5      obj.o      cwh5.cpp
Total files(directors) = 3.


./OSWorks/cwh1:
cwh1      obj.o      cwh1.cpp
Total files(directors) = 3.


./OSWorks/cwh2:
cwh2.cpp      obj.o      cwh2
Total files(directors) = 3.


./OSWorks/cwh4:
cwh4.cpp      obj.o      cwh4
Total files(directors) = 3.


./OSWorks/cwh6:
obj.o      p6.cpp      cwh6
Total files(directors) = 3.


./OSWorks/cwh3:
file1      obj.o      cwh3.cpp      cwh3
```

3.总结

这个上机实践无论从知识上还是心理上都给了我很大的收获。上了半个学期 Linux 操作系统课，除了平时在机房跟听老师的讲解之后做一些之外，回到宿舍就很少自己动手做。所以在做实验之前一想到这个 Linux 课程设计，心里面就很发慌，因为自己手生，因为题目里有很多自己没有接触过的知识点，因此一直觉得这个实验是一个很大的心病。到做的时候到处碰壁，很受打击，信心几乎被磨灭，但是还是通过查阅大量文献和资料完成了任务。

在这期间我也遇到了很多问题，比如在最开始时，我虽然通过了解 Linux 的一些掉用命

令的原理完成了代码，但是却不知道该如何把代码转换成命令的形式，这让我花费了很多时间在网络上学习，最后成功将编译好的可执行文件添加进了环境变量成功实现了目标。

做这个实验，很大程度上都要执着，否则有可能做不下去。这个实验让我从一个初学者到什么都不用看的熟练，所以我想我有收获了。在做实验过程中我比很多人少走弯路，我觉得这和认真有关，先看完是要步骤再做会更加容易一些，如果看一步做一步根本就不知道自己做那一步是为了实现什么目标。

四、参考文献

[1] Sriranga Veeraraghavan, 卢涛. 精通 shell 编程[M]. 人民邮电出版社, 2003.

[2] 敬茂华. Linux(上)的 Shell 编程[J]. 软件世界, 2000(12):102-103.

[3] 杨俊, YANGJun. Windows Shell 编程机制分析及应用研究[J]. 现代计算机, 2007(3).

[4] MarkG.Sobell, 索贝尔, 杨明军,等. Linux 命令、编辑器与 Shell 编程[M]. 清华大学出版社, 2007.

[5] 邓天权, 冯平兴, 陈洪彬. Linux 命令应用详解[J]. 2009.

[6] 施威铭研究室. Linux 指令参考手册[M]. 中国青年出版社, 2000.

[7] Reese W . Nginx: The high-performance web server and reverse proxy[J]. Linux Journal, 2008.

[8] https://kknews.cc/code/ybyqeeb.html

[9] https://www.cnblogs.com/wanghao-boke/p/11959330.html

[10] https://blog.csdn.net/haovin/article/details/92782985

[11]https://blog.csdn.net/weixin_43428283/article/details/107133649?utm_medium=distribute.pc_relevant_bbs_down.none-task-blog-baidujs-1.nonecase&depth_1-utm_source=distribute.pc_relevant_bbs_down.none-task-blog-baidujs-1.nonecase

[12]https://stackoverflow.com/questions/6473766/syntax-error-near-unexpected-token-in-r