

Lab03b Assignment

班级: 192112
学号: 19373073
姓名: 何潇龙

0. (选做, 不算分, 不限语言) 批量删除 IDEA 工程文件

参考如下的目录结构:

```
./sp_java/  
├── .idea  
│   ├── .gitignore  
│   ├── misc.xml  
│   ├── modules.xml  
│   └── workspace.xml  
├── sp_java.iml  
└── src  
    ├── Test.class  
    └── Test.java
```

在当前目录下可能存在多个类似于如上 `sp_java` 工程的文件夹。

要求批量删除这些文件夹中的 IDEA 工程文件, 转换成如下的干净的目录结构。

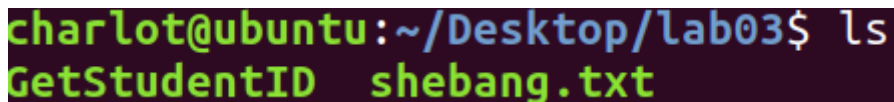
```
./sp_java  
└── src  
    └── Test.java
```

1. shebang

- 假如在脚本的第一行放入 `#!/bin/rm` 或者在普通文本文件中第一行放置 `#!/bin/more`, 然后将文件设为可执行权限执行, 看看会发生什么, 并解释为什么。
- `#!/bin/rm`

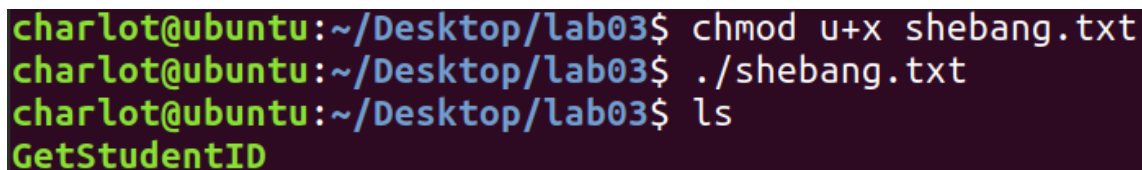
描述发生了什么(截图)

先将 `#!/bin/rm` 插入到脚本第一行, 查看目录中的文件 (实验文件为 `shebang.txt`)



```
charlot@ubuntu:~/Desktop/lab03$ ls  
GetStudentID  shebang.txt
```

赋予执行权限再执行, 再次查看目录中的文件, 发现 `shebang.txt` 消失了



```
charlot@ubuntu:~/Desktop/lab03$ chmod u+x shebang.txt  
charlot@ubuntu:~/Desktop/lab03$ ./shebang.txt  
charlot@ubuntu:~/Desktop/lab03$ ls  
GetStudentID
```

为什么

因为脚本中的第一句意思为执行 `bin` 中的 `rm`, `rm` 的作用是删除文件, 所以文件被删除

- `#!/bin/more`

描述发生了什么(截图)

文件内容被打出来了

```
charlot@ubuntu:~/Desktop/lab03$ chmod u+x shebang.txt
charlot@ubuntu:~/Desktop/lab03$ ./shebang.txt
#!/bin/more
This is shebang.txt
```

为什么

和上一题同理，也是执行bin/后的命令，more用于分页显示文件内容，这里只有两行因此全显示了出来。

2. 运行、打印

- 编写一个 bash 脚本，执行该脚本文件将得到两行输出，第一行是你的学号，第二行是当前的日期（考虑使用 `date` 命令）。对该脚本文件的要求是
 - 文件名为 `date- $\{\text{你的学号}\}$` ，比如 `date-15131049`
 - 用户可以在任意位置只需要输入文件名就可以执行该脚本文件
 - 不破坏除用户家目录之外的任何目录结构，即不要在家目录之外的任何地方增删改任何文件请详细叙述你的操作过程以及操作过程的截图，并给出你所编写的脚本文件的代码。

首先使用vi编辑器编辑脚本

```
#!/bin/bash
echo "19373073"
echo $(date)
```

之后在bashrc文件中追加脚本的地址进入环境变量中，这样就可以在各个地方打开该文件。

```
charlot@ubuntu:~
File Edit View Search Terminal Help

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
export PATH=$PATH:/home/charlot/Desktop/lab03
~
~
".bashrc" 118L, 3817C                               118,18      Bot
```

之后再执行：

```
charlot@ubuntu:~$ date-19373073
19373073
Thu Apr 22 02:45:57 PDT 2021
```

3. 正则

- 完成[LeetCode: 193. Valid Phone Numbers](#) | [leetcode-cn](#) (体会 `grep -E`, `grep -P`, `egrep`, `awk` 等的差异)

```
grep -P '^[0-9]{3}-|\\([0-9]{3}\\) [0-9]{3}-[0-9]{4}$' file.txt
```

AC 截图:

力扣 学习 题库 讨论 竞赛 求职 商店

新功能 下载 App 会员 中 智能模式

题目描述 评论 (176) 题解 (120) 提交记录

执行结果: 通过 显示详情

执行用时: 4 ms, 在所有 Bash 提交中击败了 62.41% 的用户

内存消耗: 3 MB, 在所有 Bash 提交中击败了 94.94% 的用户

炫耀一下:

写题解, 分享我的解题思路

提交时间	提交结果	运行时间	内存消耗	语言
几秒前	通过	4 ms	3 MB	Bash

题目列表 随机一题 193/2058 下一题 贡献 提交

4. 文件读入

- 完成[LeetCode: 195. Tenth Line](#) | [leetcode-cn](#), 给出你的代码和 AC 截图。(提示: [怎么读取每一行](#))

```
i=0
cat file.txt | while read line
do
    i=$((i+1))
    if [ $i -eq 10 ]; then
        echo $line
    fi
done
```

AC 截图:

LeetCode

Day 22

Explore

Problems

Mock

Contest

Discuss

Store

We're hiring. apply today!

Premium

Description

Solution

Discuss (127)

Submissions

Bash

Success

Details >

Runtime: 12 ms, faster than 74.74% of Bash online submission.

Line.

Memory Usage: 3.6 MB, less than 33.80% of Bash online submission.

Tenth Line.

Next challenges:

Word Frequency

Valid Phone Numbers

Transpose File

Show off your acceptance:

f

t

in

Time Submitted	Status	Runtime	Memory
04/22/2021 18:01	Accepted	12 ms	3.6 MB
04/22/2021 18:01	Accepted	16 ms	3.5 MB
04/22/2021 18:01	Accepted	12 ms	3.6 MB

1

2

3

4

5

6

7

8

9

Read from the file file.txt and output the tenth line to stdout.

i=0

cat file.txt | while read line

do

i=\$((i+1))

if [\$i -eq 10]; then

echo \$line

fi

done

Your previous code was restored from your local storage. [Reset to default](#)

X

X

<

195/1836

>

Contribute i

Submit

5. 语法

- 完成一个简单的交互设计，根据用户输入输出对应内容，具体交互内容随意，要求至少用上 `select`，`case` 和 `read`。

```
#!/bin/bash
echo "请选择你想买的手机"
read phone
case $phone in
    Apple | "苹果" ) echo "你选择了苹果手机";;
    Samsung | "三星" ) echo "你选择了三星手机";;
    Huawei | "华为" ) echo "你选择了华为手机";;
    Xiaomi | "小米" ) echo "你选择了小米手机";;
    * ) echo "不好意思本店没有这种手机"
esac
echo "请选择你想要的内存大小的序号"
select size in "64G" "128G" "256G"
do
    echo "你选择了 $size 的手机"
    break
done
exit 0
```

交互体验截图：

```
charlot@ubuntu:~/Desktop/lab03$ ./Q5.sh
请选择你想买的手机
Apple
你选择了苹果手机
请选择你想要的内存大小的序号
1) 64G
2) 128G
3) 256G
#? 3
你选择了 256G 的手机
```

6. 综合实验

- 编写 Shell 脚本 `addowner.sh` 将某目录下面所有的文件名后面加上文件所有者的名字。比如 `a.txt` 和 `file` 的所有者都为 `owner`，文件名修改后分别为 `a[owner].txt` 和 `file[owner]`。
 - 使用用法： `./addowner 目录名称`。（若无目录名称这一参数，则默认为当前目录）
 - 对于子目录，名称不变。
 - （提示：为了测试效果，请通过 `useradd` 创建若干用户，并可通过 `chown` 改变文件所有者。另外，网上的答案是错的。）

```
#!/bin/bash
if [ $# -eq 1 ]; then
    cd $1
fi
for filename in `ls`
do
    if [ ! -d $filename ];then
        List=(`ls -l $filename`)
        Owner=${List[2]}
        prefix=${filename%.*}
        suffix=${filename##*.}
        if [ $filename = $prefix ]; then
            mv $filename "$filename[$Owner]"
        else
            mv $filename "$prefix[$Owner].$suffix"
        fi
    fi
done
```

运行结果截图(包含目录、文件(含 `.` 与不含 `.`)，并体现出多个文件所有者)：

先准备好测试文件，给不同文件分配不同拥有者

```

charlot@ubuntu:~/Desktop/test$ sudo useradd AA
charlot@ubuntu:~/Desktop/test$ sudo useradd BB
charlot@ubuntu:~/Desktop/test$ sudo chown AA abc
charlot@ubuntu:~/Desktop/test$ sudo chown BB abc.aa
charlot@ubuntu:~/Desktop/test$ ls -l
total 16
-rw-r--r-- 1 charlot charlot 400 Apr 25 07:47 aa.abccc
-rwxrwxrwx 1 AA      charlot 398 Apr 25 07:50 abc
-rwxrwxrwx 1 BB      charlot 398 Apr 25 07:50 abc.aa
-rwxrwxrwx 1 charlot charlot 350 Apr 25 08:00 addowner

```

再测试函数

```

charlot@ubuntu:~/Desktop/test$ ls -l
total 20
-rw-r--r-- 1 charlot charlot 400 Apr 25 07:47 aa.abccc
-rwxrwxrwx 1 AA      charlot 398 Apr 25 07:50 abc
-rwxrwxrwx 1 BB      charlot 398 Apr 25 07:50 abc.aa
drwxr-xr-x 2 charlot charlot 4096 Apr 25 08:08 balabala
-rwxrwxrwx 1 charlot charlot 350 Apr 25 08:00 hahaha
charlot@ubuntu:~/Desktop/test$ cd /home/charlot/Desktop/lab03
charlot@ubuntu:~/Desktop/lab03$ ./addowner /home/charlot/Desktop/test
charlot@ubuntu:~/Desktop/lab03$ cd /home/charlot/Desktop/test
charlot@ubuntu:~/Desktop/test$ ls -l
total 20
-rw-r--r-- 1 charlot charlot 400 Apr 25 07:47 'aa[charlot].abccc'
-rwxrwxrwx 1 AA      charlot 398 Apr 25 07:50 'abc[AA]'
-rwxrwxrwx 1 BB      charlot 398 Apr 25 07:50 'abc[BB].aa'
drwxr-xr-x 2 charlot charlot 4096 Apr 25 08:08 balabala
-rwxrwxrwx 1 charlot charlot 350 Apr 25 08:00 'hahaha[charlot]'

```

完成本综合实验的过程和体会：

完成这个实验借鉴了网上一些“批量命名”的思路，然后在重命名的时候遇到了一些困难，因为还是不大熟悉正则表达式所以用了很原始的方法，把"."前后的内容拆开再追加用户名。

7. 实验感想

这次实验操作性特别强，不仅熟悉了shell脚本的编写，而且学到了很多课本上没有学到的知识，比如正则表达式。看到自己写的脚本能有实际的效果还挺有成就感的。