浙江大学实验报告

课程名称：Linux 应用技术基础 实验类型：综合型

实验项目名称：实验 2 程序设计

学生姓名：王涵 专业：电子信息工程 学号：3200104515

电子邮件地址：2637148441@qq.com

实验日期：2021 年 12 月 29 日

一、 实验环境

```
wh@wh-virtual-machine:~$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 126
model name      : Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz
stepping        : 5
microcode       : 0x70
cpu MHz         : 1190.401
cache size      : 6144 KB
physical id     : 0
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 27
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmo
n nopl xtopology tsc_reliable nonstop_tsc cpuid pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor
lahf_lm abm 3dnowprefetch cpuid_fault invpcid_single ssbd ibrs ibpb stibp ibrs_enhanced fsgsbase tsc_adjust bmi1 avx2 smep bmi2 invpcid avx512f avx512dq rdseed adx smap clf
lushopt avx512cd sha_ni avx512bw avx512vl xsaveopt xsavec xsaves arat pku ospke avx512_vpopcntdq md_clear flush_l1d arch_capabilities
bugs            : spectre_v1 spectre_v2 spec_store_bypass swapgs itlb_multihit
bogomips        : 2380.80
clflush size    : 64
cache_alignment : 64
address sizes   : 43 bits physical, 48 bits virtual
power management:
```

计算机配置见上，操作系统环境 windows10，安装 VMware 虚拟机，Linux 版本为 Ubuntu20.04 版本

//填写你的计算机配置，操作系统环境，Linux 版本

二、 实验内容和结果及分析

实验一：

```
wh@wh-virtual-machine:~/professional/courses$ ./1.sh file
wh
11:36
wh@wh-virtual-machine:~/professional/courses$ ./1.sh file file
输入的参数过多
wh@wh-virtual-machine:~/professional/courses$ ./1.sh major
wh@wh-virtual-machine:~/professional/courses$ ls -l
总用量 12
-rwxrwxrw- 1 wh wh  463 12月 12 12:15 1.sh
-rw-rw-r-- 1 wh wh    0 12月 12 11:36 file
drwxrwxr-x 2 wh wh 4096 11月  3 18:52 general
drwxrwxr-x 5 wh wh 4096 11月  3 18:53 major
```

```
# 程序名:1.sh
# 作者：王涵　学号：3200104515
#! /bin/bash
if [[ $#>=2 ]] # 判断输入参数的数量，若太多则输出 输入参数过多并退出
then
        echo "输入的参数过多"
        exit 1
elif [[ $#==1 ]] # 如果只输入了一个命令行参数
then
        if [ -f $1 ] # 判断该文件是否为普通文件
        then
                set -- $(ls $1 -l) # 使用 set 修改位置参数的值
                echo $3 # 返回第三个参数
                echo $8
        fi
        exit 0 # 退出程序
elif [[ $#==0   ]]
then
        echo "没有输入参数"
        exit 0
fi
```

实验二：

```
wh@wh-virtual-machine:~/professional/courses$ ./2.sh .
普通文件数目:6
子目录文件数目:9
可执行文件的数目: 2
普通文件字节数总和: 69
```

```
# 程序名：2.sh
# 作者：王涵　学号：3200104515
#！ /bin/bash/
echo "普通文件数目:$(find $1 -type f|wc -l)"
echo "子目录文件数目:$(find $1 -type d|wc -l)"
echo "可执行文件的数目：$( ls $1 -l |grep '^-..x'|wc -l)"
echo "普通文件字节数总和：$(find $1 -type f|wc -c)"
```

实验三：

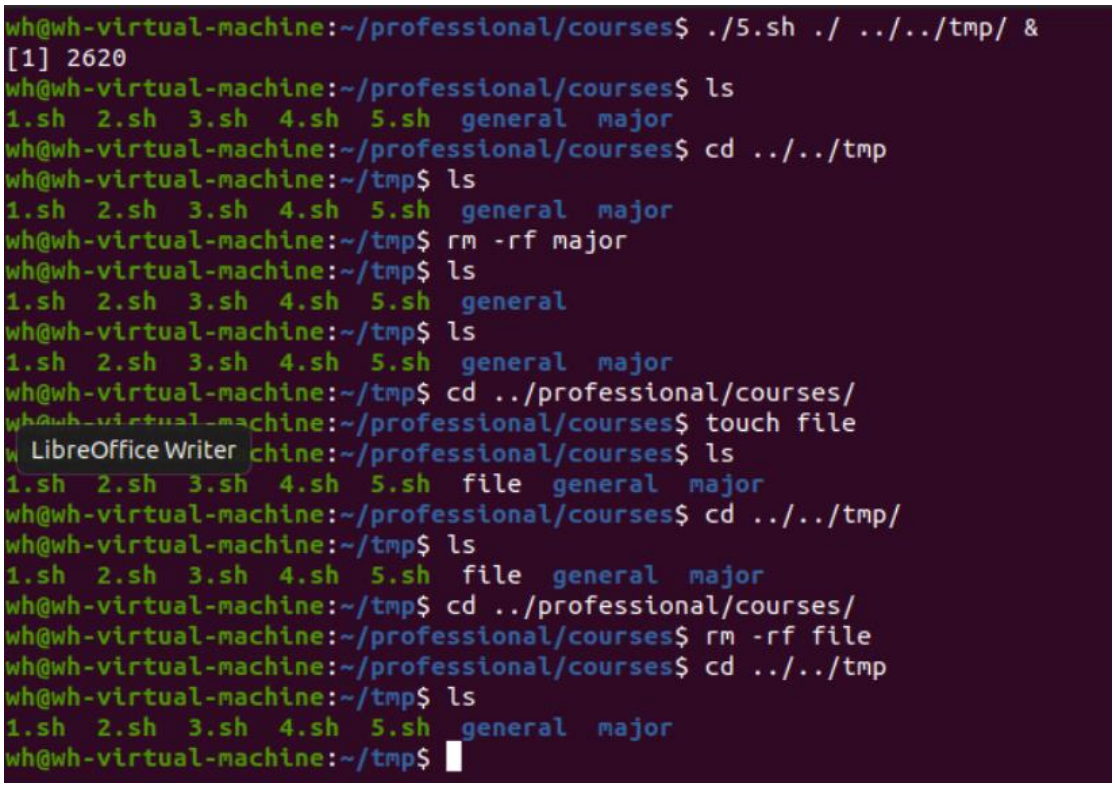
```
wh@wh-virtual-machine:~/professional/courses$ ./3.sh ass123124124124512a
assa
a
a
s
s
success
wh@wh-virtual-machine:~/professional/courses$ ./3.sh assf123124124124512fa
assffa
a
a
f
s
fail
wh@wh-virtual-machine:~/professional/courses$ ./3.sh a123412a41412f12412a214a
aafaa
a
a
a
a
f
f
success
wh@wh-virtual-machine:~/professional/courses$ ./3.sh asdfdsa
asdfdsa
a
a
s
s
d
d
f
f
success
```

```
# 程序名：3.sh
# 作者：王涵 学号：3200104515
#! /bash/bin
tmp=0 #tmp 为字母串长度
len=$(echo $1|wc -c) #获得输入字符串的长度
a=1
while (( a<len ))
do
    b=$(echo $1|cut -c $a) #取出每个字符
    if [[ $b>='a' && $b<='z' ]] || [[ $b>='A' && $b<='Z' ]] #如果为字母
    then
        char=$char$b #添加到字母串中
        tmp=$[$tmp+1]
    fi
    a=$[$a+1]
done
echo $char
i=1
j=$tmp
flag=0 #标志是否 break
while (( i<=j )) #j 从后，i 从前往中间遍历
do
    front=$(echo $char|cut -c $i) #前面的字母
    end=$(echo $char|cut -c $j) #后面的字母

    echo $end
    echo $front

    if [ $front != $end ] #如果两个字母不同
    then
        echo "fail"
        flag=1 #修改标志
        break
    fi
    i=$[$i+1]
    j=$[$j-1]
done

if [ $flag -ne 1 ] #如果循环不是从 break 中跳出
then
    echo success #echo success 代表是回文串
fi
```

实验四：

```
# 程序名：4.sh
# 作者：王涵　学号：3200104515
#! /bash/bin
for x in $(find ./ -size +100k -type f) #  循环大小大于 100k 的文件
do
        cp $x ~/tmp/ #进行 copy 备份
        echo success #如果成功则输出 success
done
```

实验五：


wh@wh-virtual-machine:~/professional/courses$ ./5.sh ./ ../../tmp/ &
[1] 2620
wh@wh-virtual-machine:~/professional/courses$ ls
1.sh 2.sh 3.sh 4.sh 5.sh general major
wh@wh-virtual-machine:~/professional/courses$ cd ../../tmp
wh@wh-virtual-machine:~/tmp$ ls
1.sh 2.sh 3.sh 4.sh 5.sh general major
wh@wh-virtual-machine:~/tmp$ rm -rf major
wh@wh-virtual-machine:~/tmp$ ls
1.sh 2.sh 3.sh 4.sh 5.sh general
wh@wh-virtual-machine:~/tmp$ ls
1.sh 2.sh 3.sh 4.sh 5.sh general major
wh@wh-virtual-machine:~/tmp$ cd ../professional/courses/
wh@wh-virtual-machine:~/professional/courses$ touch file
LibreOffice Writer
chine:~/professional/courses$ ls
1.sh 2.sh 3.sh 4.sh 5.sh file general major
wh@wh-virtual-machine:~/professional/courses$ cd ../../tmp/
wh@wh-virtual-machine:~/tmp$ ls
1.sh 2.sh 3.sh 4.sh 5.sh file general major
wh@wh-virtual-machine:~/tmp$ cd ../professional/courses/
wh@wh-virtual-machine:~/professional/courses$ rm -rf file
wh@wh-virtual-machine:~/professional/courses$ cd ../../tmp
wh@wh-virtual-machine:~/tmp$ ls
1.sh 2.sh 3.sh 4.sh 5.sh general major
wh@wh-virtual-machine:~/tmp$

```
# 程序名：5.sh
# 作者：王涵 学号：3200104515
#! /bash/bin
ini=$1 #原地址
target=$2 #目标地址

for x in $(ls $ini)   #备份操作
do
    set -- $(ls $x -l)
    if [[ $1==^d ]] #若为目录文件
    then
        cp -r $ini$x $target$x
    else #若为普通文件
        cp $ini$x $target$x
    fi
done
while ((1))   #同步操作
do
    for x in $(ls $ini)   #从原地址向目标地址的同步
    do
        if [ -f $x ]
        then
            cp -u $ini$x $target
        elif [ -d $x ]
        then
            cp -u -r $ini$x $target
        fi
    done
    for x in $(ls $target)   #从目标地址向原地址的同步
    do
        if [ -f $x ]
        then
            cp -u $target$x $ini
        elif [ -d $x ]
        then
            cp -r -u $target$x $ini
        fi
    done
    for x in $(ls $target) #  检查原地址文件是否被删除
    do
        newfile=$(ls $ini|grep $x)
        if [ -z $newfile ] #若字符串为空 代表原地址文件已被删除
        then
            rm -rf $target$x #将目标文件进行删除
        fi
    done
    sleep 10 #sleep 之后进行下一次的同步操作
done
```

三、　讨论、心得（必填）（10 分）

在本次实验中，两次写实验相隔时间较久，因此重新时遇到了很多困难，比如忘记[]与[[]]与(())的区别等；其中实验三，在我使用了 rev 和 tr 完成了要求之后才看到题目要求不能使用 rev 与 tr，这又让我重新来过，因为需要必要的正则表达式判断，而如何书写正则表达式又被我忘了差不多，所以在写脚本的过程中有些坎坷；最后同步操作的实验，在我仔细思考完可以用过比较 stat 命令中的修改时间来判断更新文件且书写完代码之后，无意看到可以使用 cp -u 实现一步到位，这让我又学到了一点新的知识；其实，在使用很多命令之前，我并不知道 linux 系统拥有这方面功能的命令，所以绕了很多弯路，不过也了解到了很多命令的别样使用；实验三刚开始想用 for 循环直接取出每个字符，但遇到了困难，一直没办法单独取出单个字符进行判断，直到知道了 cut 命令，才成功的取出了单个字符，之后等于的判断也遇到了问题，究其原因是我还不了解各种括号区别，最后在百度之后才成功完成了实验三