**实验8 Shell应用**

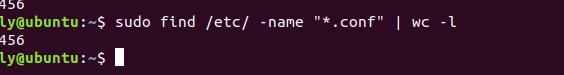
实验目的

1. 熟悉掌握shell正则表达式的使用
2. 掌握grep、awk和sed命令的常用操作
3. 能够熟练使用shell的文本处理命令编写应用脚本

实验内容

1. **shell文本处理命令（自己写代码，上传命令和运行截图）**
2. 写一个shell命令，实现查找/etc/目录下（包括子目录）以.conf结尾的文件个数；

sudo find /etc/ -name "\*.conf" | wc -l



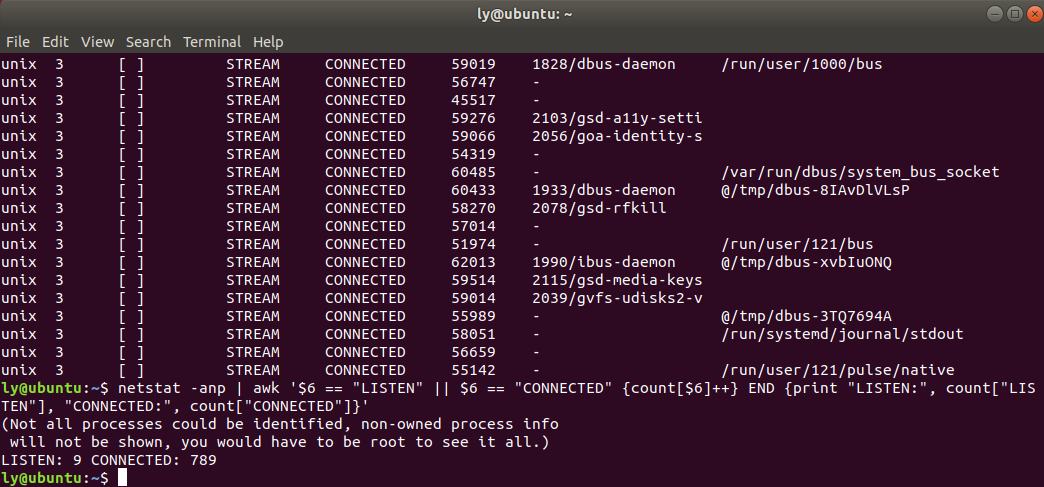
1. 写一个shell命令，实现统计系统中可登录的用户（/etc/passwd文件中以/bin/bash结尾的用户）有多少；

sudo grep '/bin/bash$' /etc/passwd | wc -l



1. 写一个shell命令，实现统计netstat -anp命令执行中，状态为LISTEN和CONNECTED连接数各是多少（awk模式匹配）$1~/root/

netstat -anp | awk '$6 == "LISTEN" || $6 == "CONNECTED" {count[$6]++} END {print "LISTEN:", count["LISTEN"], "CONNECTED:", count["CONNECTED"]}'



4.写一个shell命令，输出一个文本文件file.txt（自己编辑该文件）中空行的行数。

假定file.txt文件内容为：^$

hello

空行

world

Hello

空行

空行

Shell

空行

图形用户界面, 应用程序

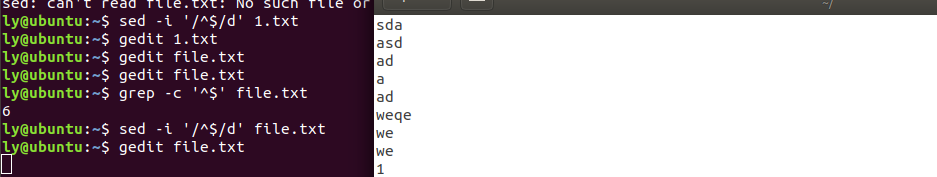
描述已自动生成

grep -c '^$' file.txt



5.一个shell命令，删除file.txt中的空行，并保存删除后的该文件。

sed -i '/^$/d' 1.txt



**二、shell脚本编写（自己写代码，上传代码文件和运行截图）**

1.写一个脚本文件，实现四则运算计算器功能（+-\*/），要求：输入参数：如20+30，输出运算结果：50（可以分别读入运算数1、运算符、运算数2）read -p s

#!/bin/bash

read -p "请输入算术表达式（如20+30）：" expression

num1=$(echo "$expression" | cut -d'+' -f1 | cut -d'-' -f1 | cut -d'\*' -f1 | cut -d'/' -f1)

operator=$(echo "$expression" | sed 's/[0-9]\*//g')

num2=$(echo "$expression" | cut -d"$operator" -f2)

case $operator in

+) result=$(($num1 + $num2));;

-) result=$(($num1 - $num2));;

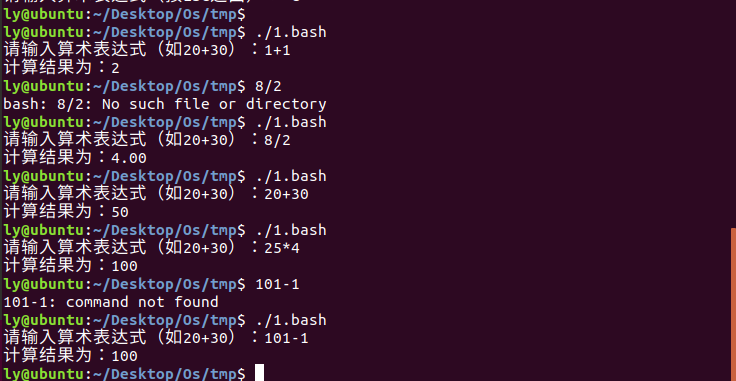
\\*) result=$(($num1 \* $num2));;

/) result=$(echo "scale=2; $num1 / $num2" | bc);;

\*) echo "运算符不支持，请重新输入"; exit 1;;

esac

echo "计算结果为：$result"



2. 写一个脚本，根据用户名查询用户的所有信息（UID、组、组ID、家目录，是否有登录权限最后一个字段值为/bin/bash）

cat /etc/passwd |awk -F: '$1~/'^$s$'/{print $3}'

#!/bin/bash

read -p "请输入用户名：" username

user\_info=$(cat /etc/passwd | grep "^$username:")

if [ -z "$user\_info" ]; then

echo "用户 $username 不存在。"

exit 1

fi

uid=$(echo "$user\_info" | awk -F: '{print $3}')

group=$(echo "$user\_info" | awk -F: '{print $4}')

gid=$(echo "$user\_info" | awk -F: '{print $4}' | cut -d, -f1)

home\_dir=$(echo "$user\_info" | awk -F: '{print $6}')

login\_shell=$(echo "$user\_info" | awk -F: '{print $NF}')

if [ "$login\_shell" = "/bin/bash" ]; then

login\_status="是"

else

login\_status="否"

fi

echo "用户 $username 的信息如下："

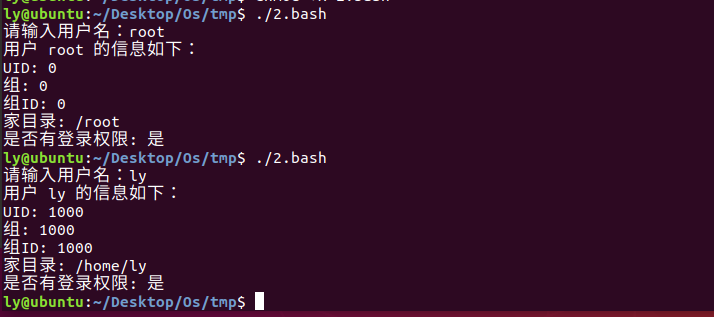
echo "UID: $uid"

echo "组: $group"

echo "组ID: $gid"

echo "家目录: $home\_dir"

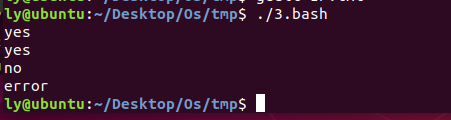
echo "是否有登录权限: $login\_status"



3.写一个脚本统计文件IP.txt中的每一行是否是正确的IP地址。如果是正确的IP地址输出：yes；如果是错误的IP地址，四段号码的话输出：no，否则的话输出：error；假设IP.txt内容为：

192.168.1.1  
192.168.1.0  
300.0.0.0  
123形状, 矩形

描述已自动生成



4.（选做）写一个脚本，根据进程ID，输出进程的状态和占用资源情况。（awk分析ps aux命令的输出信息）

#!/bin/bash

read -p "请输入进程ID：" pid

process\_info=$(ps aux | awk -v pid="$pid" '$2 == pid {print}')

if [ -z "$process\_info" ]; then

echo "未找到进程ID为 $pid 的进程。"

exit 1

fi

status=$(echo "$process\_info" | awk '{print $8}')

cpu=$(echo "$process\_info" | awk '{print $3}')

mem=$(echo "$process\_info" | awk '{print $4}')

command=$(echo "$process\_info" | awk '{print $11}')

echo "进程ID为 $pid 的进程状态和资源占用情况："

echo "状态：$status"

echo "CPU占用：$cpu%"

echo "内存占用：$mem%"

echo "命令：$command"

