南京邮电大学

实 验 报 告

（ 2024/2025 学年 第 一 学期 ）

课程名称	Linux 编程
实验名称	实验二：Shell 脚本编程实验
实验时间	2024 年 11 月 15 日
指导单位	计算机学院 网络空间安全系
指导教师	王磊

学生姓名	梁榆漫	班级学号	B220412/B22100106
学院(系)	计算机学院	专　业	信息安全

实 验 报 告

实验名称	Linux 实验 2			指导教师	王磊
实验类型	验证	实验学时	2	实验时间	2024.11.15

一、 实验目的和要求

通过编写和运行 Shell 脚本，掌握 Linux 环境下的脚本编程方法，提高对基本编程逻辑的理解和实际操作能力。

二、 实验环境(实验设备)

硬件：微型计算机

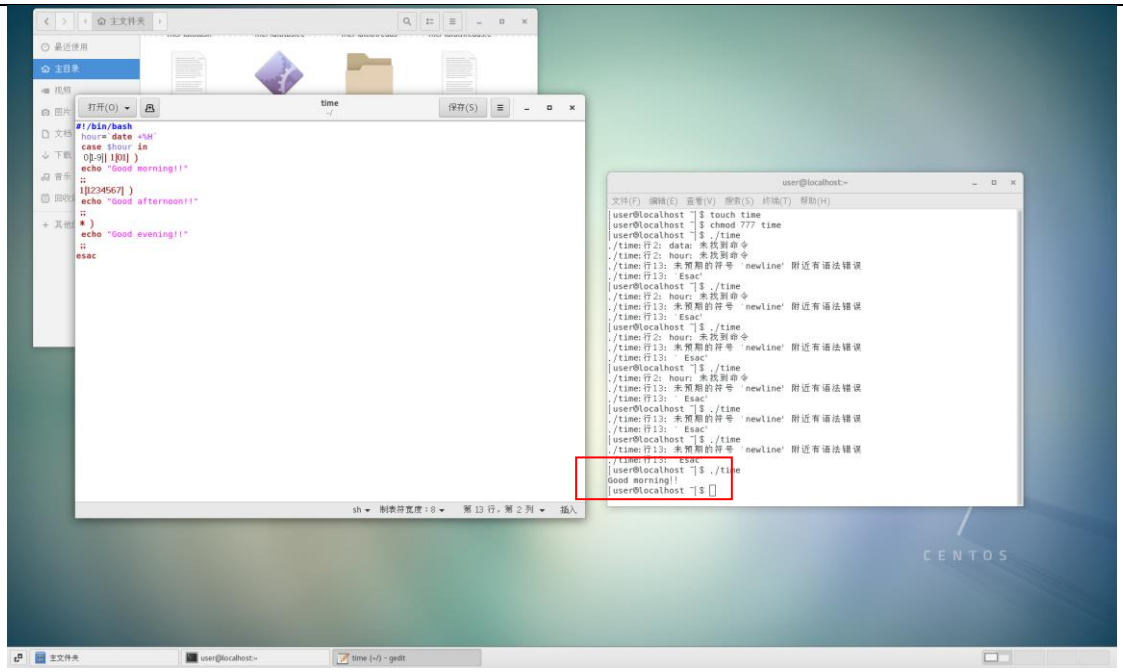
软件：实验环境：Linux 系统：CentOS 7

三、 实验原理及内容

任务 1：获取系统时间，判断是上午、下午还是晚上

脚本内容：

```
#!/bin/bash
hour=`date +%H`
case $hour in
 0[1-9] | 1[01] )
   echo "Good morning !!"
   ;;
 1[234567] )
   echo "Good afternoon !!"
   ;;
 * )
   echo "Good evening !!"
   ;;
esac
```



□ 1

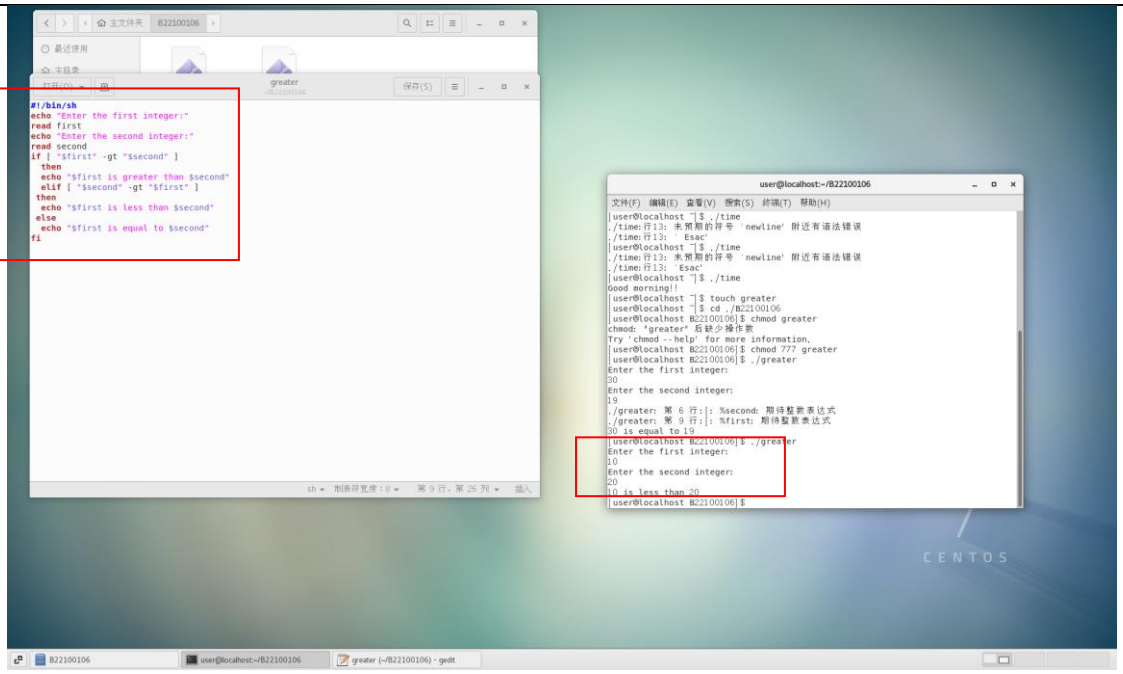
解析：

- date +%H 获取当前系统时间的小时部分（24 小时制）。
- case 语句用于多条件判断，根据 $hour 的值判断时间段：
  - 0[1-9] | 1[01]：上午（01:00-11:00）。
  - 1[234567]：下午（12:00-17:00）。
  - *：其他时间，即晚上（18:00-00:00）。
- 脚本根据条件输出对应的问候语。

任务 2：输入两个数字，判断哪个更大，并输出结果

```
#!/bin/sh
echo "Enter the first integer:"
read first
echo "Enter the second integer:"
read second
if [ "$first" -gt "$second" ]
then
  echo "$first is greater than $second"
elif [ "$first" -lt "$second" ]
then
  echo "$first is less than $second"
```

```
else
  echo "$first is equal to $second"
fi
```



□ 2

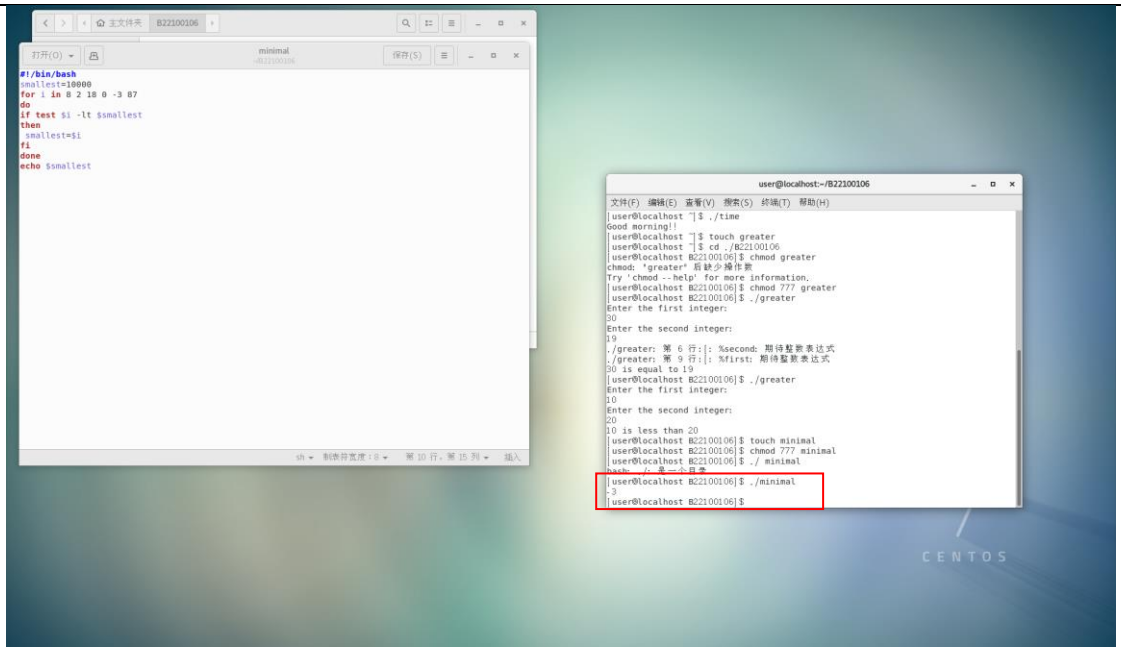
解析：

- 使用 read 命令从用户输入中读取两个整数。
- 使用条件判断 if-elif-else 比较输入的整数，-gt、-lt 分别表示“>”和“<”的判断。
- 根据比较结果输出哪一个整数更大或两者相等。

任务 3：找出给定列表中的最小值

脚本内容：

```
#!/bin/bash
smallest=10000
for i in 8 2 18 0 -3 87
do
  if test $i -lt $smallest
  then
    smallest=$i
  fi
done
echo $smallest
```



□ 3

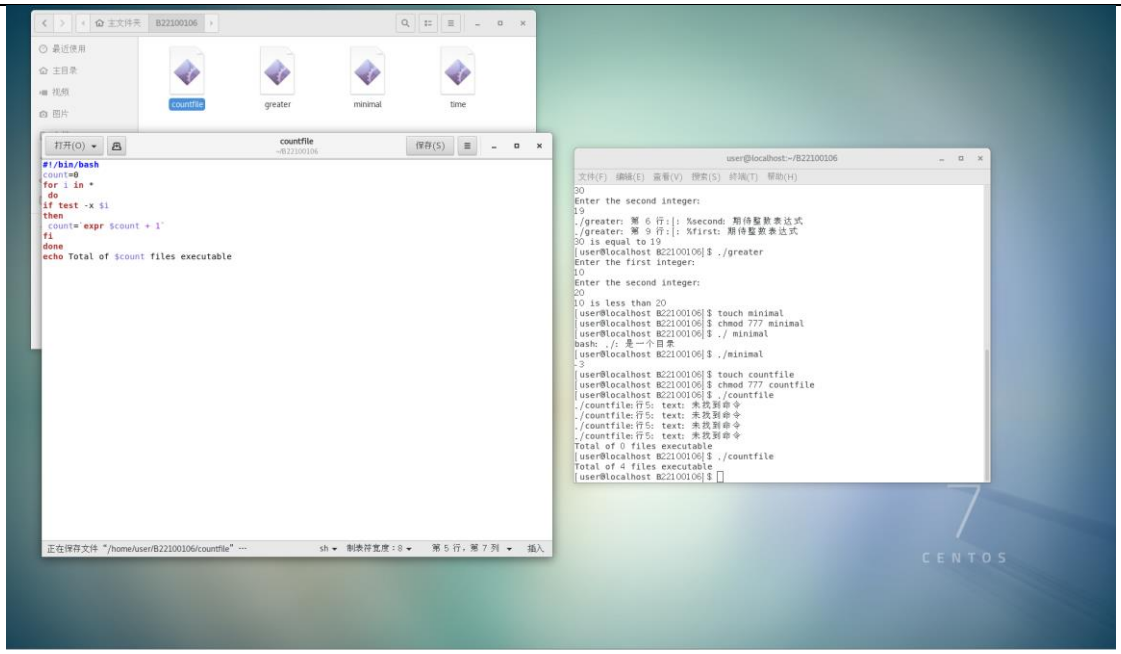
解析：

- 初始化变量 smallest 为一个较大的值（10000），用于保存当前最小值。
- 使用 for 循环遍历列表中的所有元素，每次检查是否比当前的 smallest 小。
- 使用 test 命令比较两个数的大小，若满足条件则更新 smallest。
- 脚本最终输出列表中的最小值。

任务 4：计算当前目录中可执行文件的数量

脚本内容：

```
#!/bin/bash
count=0
for i in *
do
  if test -x $i
  then
    count=`expr $count + 1`
  fi
done
echo Total of $count files executable
```



□ 4

解析：

- for i in * 遍历当前目录下的所有文件。
- test -x $i 检查文件 $i 是否具有可执行权限。
- 如果文件可执行，使用 expr 命令将计数器变量 count 加 1。
- 最后输出当前目录下所有具有可执行权限的文件总数。

任务 5：检查一个给定的数字是否为素数

脚本内容：

```
#!/bin/bash
prime() {
  flag=1
  j=2
  while [ $j -le `expr $1 / 2` ]
  do
    if [ `expr $1 % $j` -eq 0 ]
    then
      flag=0
      break
    fi
    j=`expr $j + 1`
  done
  if [ $flag -eq 1 ]
  then
    return 1
  else
    return 0
  fi
}

prime $1
if [ $? -eq 1 ]
then
  echo "$1 is a prime!"
else
  echo "$1 is not a prime!"
fi
```

□ 5

解析：

- 使用函数 prime 检查一个数字是否为素数。
- 初始值 flag=1，假设该数是素数。
- 循环从 2 开始到输入数的一半，判断是否存在因子。
    - 如果发现因子（即能被整除），将 flag 置为 0 并跳出循环。
- 根据 flag 的值返回结果：1 表示是素数，0 表示不是素数。
- 使用 $? 检查函数返回值，最终输出判断结果。

四、 实验总结

遇到的问题及解决方法

问题 1：运行脚本时出现 command not found 错误或权限不足的提示。**解决方法：**确保脚本文件保存无误，且赋予了可执行权限（使用 chmod +x 命令）。如果问题仍未解决，确认脚本的执行路径是否正确，可以使用 ./script_name.sh 或 bash script_name.sh 来运行脚本。

问题 2：在执行素数判断任务时，直接使用 ./prime 命令运行脚本，出现以下报错：

expr: 语法错误

./prime: 第 5 行:[: 2: 期待一元表达式

解决方法：该脚本需要在命令行中直接输入待判断的数字作为参数，例如：./prime 7。这样脚本才能正确获取参数并执行素数判断。实验总结

实验小结

通过本次实验，我学会了使用 Shell 脚本完成时间判断、数字比较、最小值查找、文件统计及素数判断等任务。在实验过程中，我进一步熟悉了条件判断 (if-else)、循环 (for、while) 和函数的使用。实验还让我理解了脚本的可执行权限设置的重要性，并初步体会到 Shell 编程在系统管理和任务自动化中的强大应用价值。

实 验 报 告