



实验报告

(2025 / 2026 学年 第一学期)

课程名称

Linux 编程

实验名称

实验二:

实验时间

2025 年 10 月 24 日

指导单位

计算机学院、软件学院、网络空间安全学院

指导教师

王磊

学生姓名

杨涵宇

班级学号 B23041318

学院(系)

计算机学院

专业 信息安全

实验报告

实验名称				指导教师	王磊
实验类型	上机	实验学时	4	实验时间	20

一、 实验目的和要求

掌握 Shell 脚本的基本语法和结构

熟悉 Linux 环境下 Shell 脚本的编写、执行和调试方法

掌握 Shell 脚本中变量、条件判断、循环结构、函数等的使用

能够使用 Shell 脚本解决实际的文件操作和系统管理问题

二、 实验环境(实验设备)

Ubuntu 环境、计算机

三、 实验原理及内容

任务 1：根据系统时间输出问候语

1.1 实验步骤：

创建脚本文件： nano task1.sh

输入代码：

```
#!/bin/bash
hour=$(date +\%H)
case $hour in
    0[0-9] | 1[01])
        echo "Good morning !"
        ;;
    1[2-7])
        echo "Good afternoon !"
        ;;
    *)
        echo "Good evening !"
        ;;
esac
```

保存退出： Ctrl+X → Y → Enter

添加执行权限： chmod +x task1.sh

运行脚本： ./task1.sh

1.2 实验结果：

```
sshuser@y-virtual-machine:~/桌面$ echo B23041318
B23041318
sshuser@y-virtual-machine:~/桌面$ mkdir experiment2
sshuser@y-virtual-machine:~/桌面$ cd experiment2
sshuser@y-virtual-machine:~/桌面/experiment2$ nano task1.sh
sshuser@y-virtual-machine:~/桌面/experiment2$ chmod +x task1.s
sshuser@y-virtual-machine:~/桌面/experiment2$ ls
task1.sh
sshuser@y-virtual-machine:~/桌面/experiment2$ ./task1.sh
Good evening !!
sshuser@y-virtual-machine:~/桌面/experiment2$ 
```

任务 2：比较两个整数大小

2.1 实验步骤：

创建脚本文档： nano task2.sh

输入代码：

```
#!/bin/bash
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
```

保存退出

添加执行权限： chmod +x task2.sh

运行测试：

\$./task2.sh

Enter the first integer: 1

Enter the second integer: 2

1 is less than 2

2.2 测试结果：

```
sshuser@y-virtual-machine:~/桌面/experiment2$ nano task2.sh
sshuser@y-virtual-machine:~/桌面/experiment2$ chmod +x task2.sh
sshuser@y-virtual-machine:~/桌面/experiment2$ ./task2.sh
Enter the first integer:
1
Enter the second integer:
2
1 is less than 2
sshuser@y-virtual-machine:~/桌面/experiment2$ 
```

任务 3：查找列表中的最小值

3.1 实验步骤：

创建脚本文件： nano task3.sh

```
#!/bin/bashsmallest=10000
```

```
for i in 8 2 18 0 -3 87do
```

```
    if [ $i -lt $smallest ]; then
```

```
        smallest=$i
```

```
    fidone
```

```
echo "The smallest value is: $smallest"
```

保存退出

添加执行权限： chmod +x task3.sh

运行脚本： ./task3.sh

3.2 实验结果：

输入列表： 8, 2, 18, 0, -3, 87

输出结果： The smallest value is: -3

算法验证： 手动验证列表中的最小值确实是-3

```
GNU nano 6.2                                task3.sh *
#!/bin/bash
smallest=10000

for i in 8 2 18 0 -3 87
do
    if [ $i -lt $smallest ]; then
        smallest=$i
    fi
done

echo "The smallest value is: $smallest"
```

```
sshuser@y-virtual-machine:~/桌面/experiment2$ nano task3.sh
sshuser@y-virtual-machine:~/桌面/experiment2$ chmod +x task3.sh
sshuser@y-virtual-machine:~/桌面/experiment2$ ./task3.sh
The smallest value is: -3
sshuser@y-virtual-machine:~/桌面/experiment2$
```

任务 4：统计当前目录可执行文件数量

4.1 实验步骤：

创建脚本文件： nano task4.sh

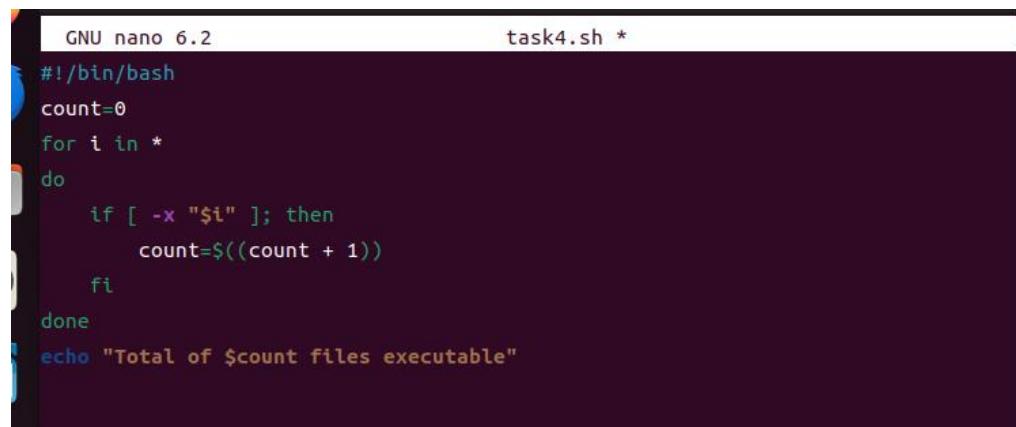
输入代码：

```
#!/bin/bash
count=0
for i in *
do
    if [ -x "$i" ]; then
        count=$((count + 1))
    fi
done
echo "Total of $count files executable"
```

保存退出

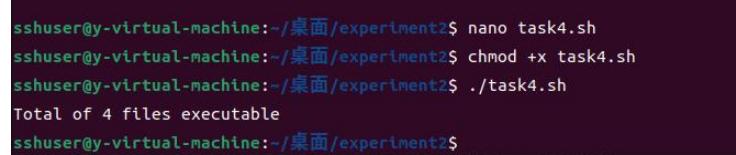
添加执行权限： chmod +x task4.sh

运行脚本： ./task4.sh



The screenshot shows a terminal window with the title "GNU nano 6.2" and the file name "task4.sh *". The code inside the file is identical to the one provided in the text above, used to count executable files in the current directory.

```
GNU nano 6.2                               task4.sh *
#!/bin/bash
count=0
for i in *
do
    if [ -x "$i" ]; then
        count=$((count + 1))
    fi
done
echo "Total of $count files executable"
```



The screenshot shows a terminal session where the user runs the script. It starts with "sshuser@y-virtual-machine:~/桌面/experiment2\$ nano task4.sh", followed by "sshuser@y-virtual-machine:~/桌面/experiment2\$ chmod +x task4.sh", and finally "sshuser@y-virtual-machine:~/桌面/experiment2\$./task4.sh". The output "Total of 4 files executable" is displayed, indicating that there are four executable files in the current directory.

```
sshuser@y-virtual-machine:~/桌面/experiment2$ nano task4.sh
sshuser@y-virtual-machine:~/桌面/experiment2$ chmod +x task4.sh
sshuser@y-virtual-machine:~/桌面/experiment2$ ./task4.sh
Total of 4 files executable
sshuser@y-virtual-machine:~/桌面/experiment2$
```

任务 5：判断素数

5.1 实验步骤：

创建脚本文件： nano task5.sh

输入代码：

```
#!/bin/bash
is_prime() {
    local num=$1
    local flag=1

    if [ $num -le 1 ]; then
        return 0
    fi
```

```

for ((j=2; j*j<=num; j++))
do
    if [ $(($num % j)) -eq 0 ]; then
        flag=0
        break
    fi
done

if [ $flag -eq 1 ]; then
    return 1
else
    return 0
fi}

if [ $# -eq 0 ]; then
    echo "Usage: ./task5.sh <number>"
    exit 1
fi

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

```

添加执行权限： chmod +x task5.sh



```

#!/bin/bash

# 定义函数
is_prime() {
    local num=$1
    local flag=1

    if [ $num -le 1 ]; then
        return 0
    fi

    for ((j=2; j*j<=num; j++))
    do
        if [ $(($num % j)) -eq 0 ]; then
            flag=0
            break
        fi
    done
}

```

```
sshuser@y-virtual-machine:~/桌面/experiment2$ ./task5.sh 17
17 is a prime!
sshuser@y-virtual-machine:~/桌面/experiment2$ ./task.sh 19
bash: ./task.sh: 没有那个文件或目录
sshuser@y-virtual-machine:~/桌面/experiment2$ ./task5.sh 19
19 is a prime!
sshuser@y-virtual-machine:~/桌面/experiment2$ ./task5.sh 199023
199023 is not a prime!
```

实验报告

四、实验小结（包括总结上机调试过程中所遇到的问题和解决方法、感想与建议等）

在本次 Shell 脚本编程实验中，我通过系统地完成五个由浅入深的编程任务，全面掌握了在 Linux 环境下进行 Shell 脚本编写与调试的实践技能。从最初简单的问候脚本到实现素数判断的函数封装，我逐步理解了 Shell 编程的核心结构，包括变量的灵活使用、多种条件判断语句的应用、循环控制的实现方法以及函数的定义与调用机制。

实验过程中，我遇到了诸如脚本执行权限不足、时间格式获取方式不当、条件判断中空格处理错误、函数返回值逻辑混淆等实际问题。通过查阅手册、反复调试和主动思考，我不仅解决了这些问题，还加深了对 Shell 脚本严谨性和灵活性的认识。尤其是在任务五中编写素数判断函数时，我不仅优化了算法效率，还学会了处理用户输入和边界情况，这使我对 Shell 脚本的健壮性和实用性有了更深的体会。此次实验让我切身体会到 Shell 脚本在系统管理、文件操作和自动化任务中不可替代的作用，同时也锻炼了我分析问题、调试代码和编写文档的综合能力，为今后从事 Linux 系统运维和信息安全相关的工作打下了坚实的实践基础。

五、指导教师评语

成 绩		批阅人		日 期	
-----	--	-----	--	-----	--