

南京邮电大学

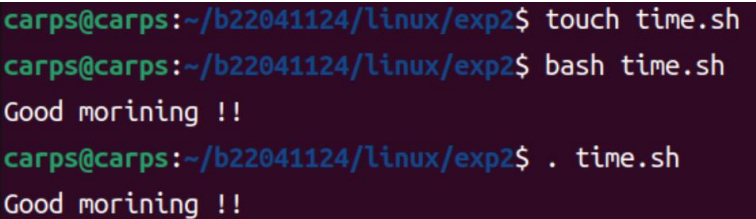
实 验 报 告

(2024/ 2025 学 年 第 一 学 期)

课程名称	Linux 编程			
实验名称	Experiment2: Shell 编程			
实验时间	2024	年 11	月 15	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	李源	班级学号	B22041124
学院(系)	计算机学院	专 业	信息安全

实 验 报 告

实验名称	Shell 编程			指导教师	王磊
实验类型	验证	实验学时	2	实验时间	2024. 11. 15
<p>一、 实验目的和要求:</p> <ol style="list-style-type: none">1. 掌握 shell 编程的相关命令2. 掌握对 shell 脚本赋执行权3. 比较 shell 和其他编程的不同					
<p>二、 实验环境(实验设备)</p> <p>硬件: 微型计算机</p> <p>软件: Ubuntu(或其他 Linux 版本)</p>					
<p>三、 实验原理及内容</p> <p>1. 获取系统时间, 检查是否在 morning, afternoon, or evening.</p> <p>(1) 输入命令:</p> <pre>#!/bin/bash hour=`date +%H` case \$hour in 0[1-9] 1[01]) echo "Good morining !!" ;; 1[2-7]) echo "Good afternoon !!" ;; *) echo "Good evening !! " ;; Esac</pre> <p>(2) 运行结果:</p> 					

(3) 原因解释:

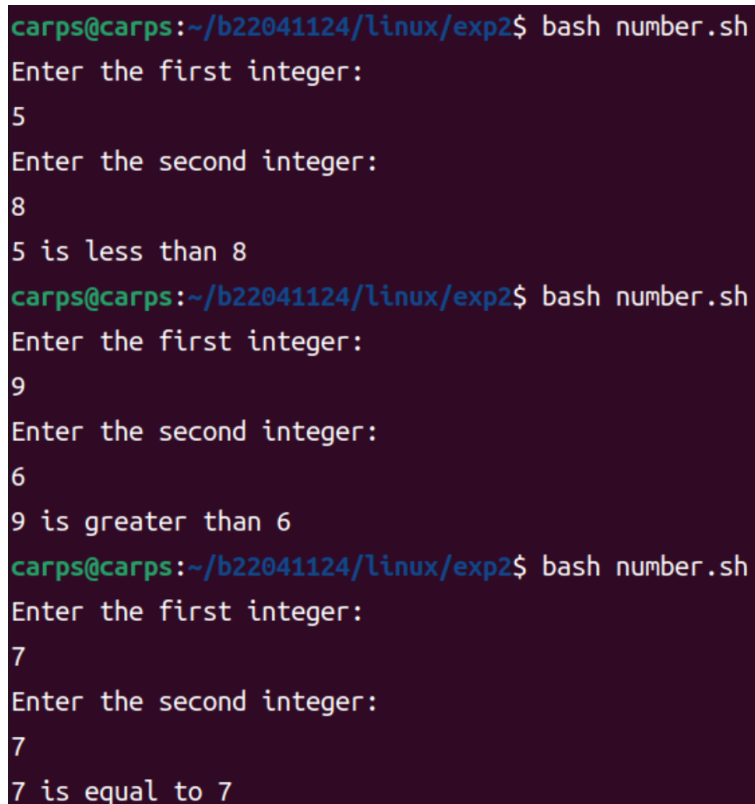
这里采用两种执行的方式,即 bash 命令和元字符“.”执行,对于 shell 代码的解释是 `hour=`date +%H``: 使用 `date` 命令获取当前的小时,并将其存储在变量 `hour` 中。`%H` 表示以两位数字形式输出小时(00 到 23), `case $hour in` 则是开始一个条件判断结构,根据变量 `hour` 的值进行匹配

2. 输入两个数, 比较哪一个更大, 输出比较结果

(1) 输入命令

```
#!/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) 运行结果:



```
carps@carps:~/b22041124/linux/exp2$ bash number.sh
Enter the first integer:
5
Enter the second integer:
8
5 is less than 8
carps@carps:~/b22041124/linux/exp2$ bash number.sh
Enter the first integer:
9
Enter the second integer:
6
9 is greater than 6
carps@carps:~/b22041124/linux/exp2$ bash number.sh
Enter the first integer:
7
Enter the second integer:
7
7 is equal to 7
```

(3) 原因解释:

输入的示例包括大于、小于和等于的情况,均可以正确判断,对于 shell 代码的解释是通过将第一个和第二个输入的数值读取并赋给 `first` 和 `second`, 比较两者大小并在屏幕打印对应的结果

3. 在所给的列表找出较小者

(1) 输入命令:

```
#!/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
```

(2) 运行结果:

```
carps@carps:~/b22041124/linux/exp2$ bash minimal.sh
-3
```

(3) 原因解释:

运行 shell 脚本, 输出的最小值为-3, 对于 shell 代码的解释, 首先最初的最小值 smallest 设置为一个较大的数 10000, 接着一个循环遍历列表, 当 i 小于 smallest 时, 将 i 的值赋给 smallest, 直到遍历结束, smallest 存储的就是列表的最小值。

4. 计算当前目录可执行文件的数量

(1) 输入命令:

```
#!/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
```

(2) 运行结果:

```
carps@carps:~/b22041124/linux/exp2$ bash filenumber.sh
Total of 3 files executable
carps@carps:~/b22041124/linux/exp2$ ls -l
总计 16
-rwxrwxrwx 1 carps carps 124 11月 15 09:28 filenumber.sh
-rwxrwxrwx 1 carps carps 123 11月 15 09:19 minimal.sh
-rwxrwxrwx 1 carps carps 292 11月 15 09:23 number.sh
-rw-rw-r-- 1 carps carps 216 12月 13 08:16 time.sh
```

```
carps@carps:~/b22041124/linux/exp2$ chmod 777 time.sh
carps@carps:~/b22041124/linux/exp2$ bash filenumber.sh
Total of 4 files executable
carps@carps:~/b22041124/linux/exp2$ ls -l
总计 16
-rwxrwxrwx 1 carps carps 124 11月 15 09:28 filenumber.sh
-rwxrwxrwx 1 carps carps 123 11月 15 09:19 minimal.sh
-rwxrwxrwx 1 carps carps 292 11月 15 09:23 number.sh
-rwxrwxrwx 1 carps carps 216 12月 13 08:16 time.sh
```

(3) 原因解释:

一开始执行 shell 脚本，发现显示有 3 个可执行文件，使用 ls -l 命令，发现实际有 4 个文件，但有执行权的文件只有 3 个；给 time.sh 赋执行权后，再次执行，数量为 4 程序也主要是通过文件是否有执行权来判断是否为可执行文件。

5. 其他管理

(1) 输入命令:

创建文件夹 others，并转到该文件夹，创建两个任意名称的文件，然后执行下列命令：

```
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

(2) 运行结果:

```
carps@carps:~/b22041124/linux/exp2$ touch prime.sh
carps@carps:~/b22041124/linux/exp2$ bash prime.sh 4
4 is not a prime!
carps@carps:~/b22041124/linux/exp2$ bash prime.sh 19
19 is a prime!
```

(3) 原因解释:

首先，初始化变量，flag 用于标记该数字是否为素数，初始值为 1（表示可能是素数），j 用于循环，从 2 开始检查。

接着，循环检查，从 2 开始，一直到 $1/2$ 的值（即输入数字的一半）。因为一个数若有因数，其因数一定小于或等于该数的一半。

然后判断是否能整除，用 `expr $1 % $j` 计算输入数字 \$1 是否能被 j 整除。如果能整除，说明 \$1 不是素数，flag 被设置为 0，并且跳出循环。每次循环结束后，将 j 增加 1，继续检查下一个数字。

如果 flag 仍然是 1，说明没有找到能整除的数，函数返回 1（表示是素数）；否则返回 0（表示不是素数）。

主程序

最后用 prime 函数，并将命令行参数 \$1 传递给它，根据返回值判断是否是素数。

实 验 报 告

四、实验小结（包括问题和解决方法、心得体会、意见与建议、实验出错信息及解决方案等）

中文五号宋体，英文五号 Times new roman 字体，1.25 倍行距

说明：这部分内容主要包括：在编程、调试或测试过程中遇到的问题及解决方法、本次实验的心得体会、进一步改进的设想等。

（一）实验中遇到的主要问题及解决方法

本次实验主要是 shell 编程，虽然在课上老师带领我们练习过，但是还是遇到了一些问题

首先就是关于文件权限的问题，问题 4 运行的结果始终比实际文件数少 1，后来经过对代码的理解发现是因为它是根据是否有执行权来判断的，因此当使用 ls -l 时发现果然有一个 shell 没有执行权，后来想到此文件是后来新建的本身就没有添加执行权，添加后没有问题

还有就是最后一个问题，直接将老师的代码复制并执行是有错误的，后来阅读代码发现，需要传递一个参数，所以对于一些验证实验一定要先理解代码，不能盲目运行。

（二）实验心得

本次实验主要是实现 shell 编程，对于一些 shell 语句的理解和如何运行 shell 脚本，以及一些文件权限内容的掌握，本次实验对于 if 的条件分支语句和 for 的循环语句都有了一定的掌握，尤其是 if 语句中条件表达式的表示，我觉得是这一次实验对于我而言最需要掌握和注意的地方。

（三）意见与建议（没有可省略）

五、支撑毕业要求指标点

☑2-1-M 能够应用数学、自然科学、计算机科学和网络空间安全的基本原理对复杂安全问题的关键过程环节进行识别和判断。

□3-2-H 能够在信息安全系统中合理的组织数据、有效存储和处理数据，正确的设计算法以及对算法进行分析和评价。

□4-1-M 针对信息安全领域复杂工程问题，能基于专业理论并采用科学方法，设计可行的实验方案，具有根据解决方案进行工程实施的能力。

六、指导教师评语

评 分 细 则	评分项	优秀	良好	中等	合格	不合格
	遵守实验室规章制度					
	学习态度					
	算法思想准备情况					
	程序设计能力					
	解决问题能力					
	课题功能实现情况					
	算法设计合理性					
	算法效能评价					
	回答问题准确度					
	报告书写认真程度					
	内容详实程度					
	文字表达熟练程度					
	其它评价意见					
	本次实验能力达成评价 (总成绩)		批阅人		日期	