

# 四川大学计算机学院、软件学院

## 实验报告

学号：\_\_\_\_\_ 姓名：\_\_\_\_\_ 专业：软件工程 班级：05 第二周

课程名称	操作系统课程设计	实验课时	2
实验项目	Shell 脚本开发	实验时间	2023/09/18
实验目的	1) 概念：Shell 编程基础 2) 演示/实践：简单的 shell 编程举例和实践		
实验环境	ARM64, MacOS, Parallels Desktop 19, Ubuntu Linux 22.04.2		

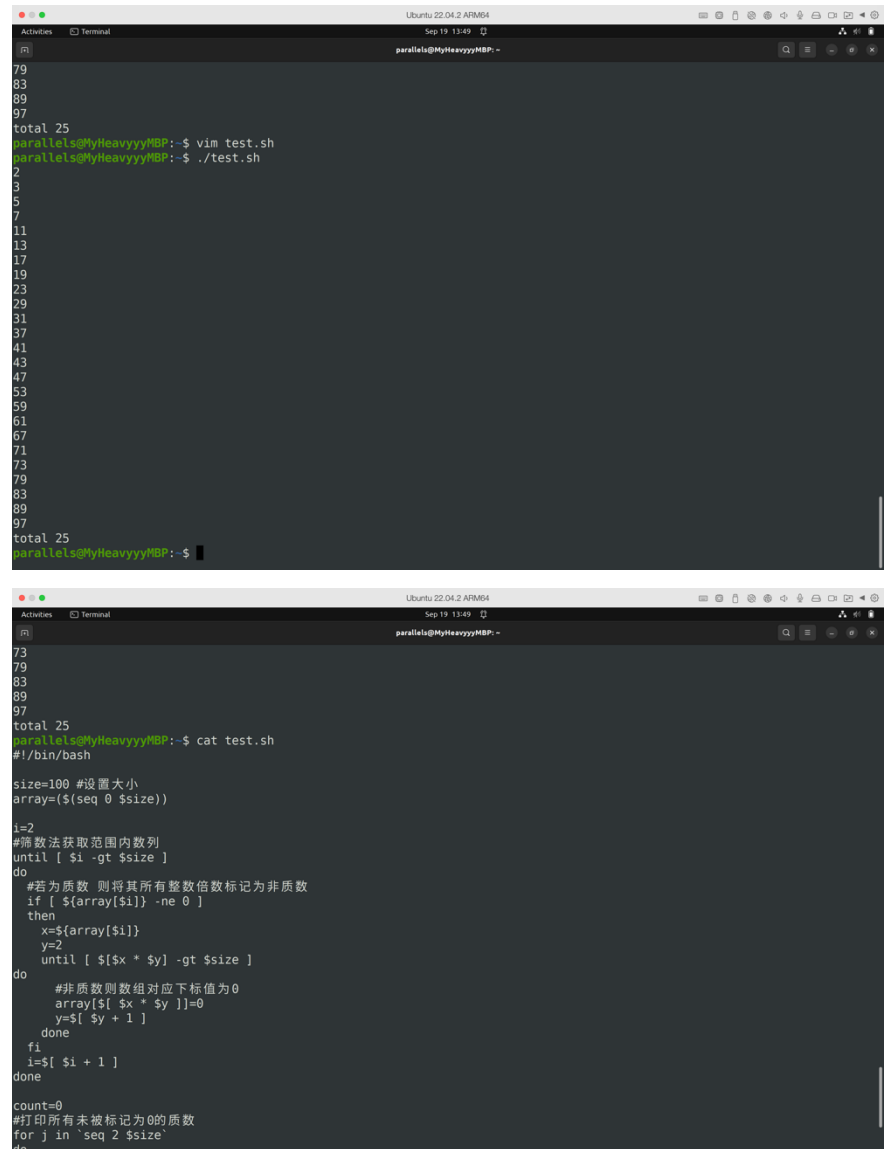
<p>实 验 内 容 (算法、程序、步骤和方法, 以及必要的截图)</p>	<p><b>Lab1: Shell 编程:</b> 运行并分析以下 shell 脚本 demo_1, 解释标注部分的含义:</p> <pre> \$cat demo_1 #!/bin/sh if [ \$# -eq 0 ]    #解释 1 如果变量 “#” 等于 0, true then     echo "Usage: \$0 ordinary_file"     exit 1 fi if [ \$# -gt 1 ]    #解释 2 如果变量 “#” 大于或等于 0, true then     echo "Usage: \$0 ordinary_file"     exit 1 fi if [ -f "\$1" ]    #解释 3 检查输入的的第一个参数是否为普通文件 then     filename="\$1"     set `ls -il \$filename`  #解释 4 设置 shell 变量 (“\$1” “\$2”...)     的新变量值为执行 “ls -il \$filename” 语句后得到的新变量: 即根据     第一个参数 (原 “\$1”、“\$filename”) 所代表的文件的索引值 (新的     “\$1”) 和详细信息们 (新的 “\$2” “\$3” ... )。     inode="\$1"     size="\$6"     echo "Name\t  Inode\t  Size"     echo     echo "\$filename\t  \$inode\t  \$size"  #解释 5 (输出结果得到的?) 输出根据旧参数所得文件的文件名、索引值、文件大小 (字节数)     exit 0 else     echo "\$0: argument must be an ordinary files"     exit 1 fi </pre> <p><b>Lab2: Shell 编程应用:</b> 寻找并打印 1 到 100 的质数和质数和。</p> <pre> #!/bin/bash  size=100 #设置大小 </pre>
---	---

```
array=$(seq 0 $size)

i=2
#筛数法获取范围内数列
until [ $i -gt $size ]
do
#若为质数 则将其所有整数倍数标记为非质数
    if [ ${array[$i]} -ne 0 ]
    then
        x=${array[$i]}
        y=2
        until [ ${x * y} -gt $size ]
        do
            #非质数则数组对应下标值为 0
            array[${x * y}]=0
            y=$((y + 1))
        done
    fi
    i=$((i + 1))
done

count=0
#打印所有未被标记为 0 的质数
for j in `seq 2 $size`
do
    if [ ${array[$j]} -ne 0 ]
    then
        echo ${array[$j]}
        count=$((count + 1))
    fi
done

#打印范围内所有质数的个数
echo total $count
```



```
79
83
89
97
total 25
parallels@MyHeavyMbp:~$ vim test.sh
parallels@MyHeavyMbp:~$ ./test.sh
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
total 25
parallels@MyHeavyMbp:~$
```

```
73
79
83
89
97
total 25
parallels@MyHeavyMbp:~$ cat test.sh
#!/bin/bash

size=100 #设置大小
array=( $(seq 0 $size) )

i=2
#筛数法获取范围内数列
until [ $i -gt $size ]
do
    #若为质数 则将其所有整数倍数标记为非质数
    if [ ${array[$i]} -ne 0 ]
    then
        x=${array[$i]}
        y=2
        until [ ${x * $y} -gt $size ]
        do
            #非质数则数组对应下标值为0
            array[${x * $y}]=0
            y=$((y + 1))
        done
    fi
    i=$((i + 1))
done

count=0
#打印所有未被标记为0的质数
for j in $(seq 2 $size)
do
```

实验结果如上图

小 结	<p>通过本实验，我掌握了：</p> <p>1) Linux Shell 的编程基础</p> <p>目前我存在的问题有：</p> <p>1) 刚开始接触 Shell 编程，仍然有些生疏</p> <p>2) 对 Shell 编程的应用场景存在疑问</p>
指导老师 评 议	<p>成绩评定：</p> <p>指导教师签名：</p>