

一、实验目的

通过编辑器完成以下 Shell 脚本，并在 Linux 系统中运行这些脚本，掌握 Shell 脚本的基础编写与执行方法，熟悉条件判断、循环、函数定义与调用等核心语法，提升 Linux 环境下的脚本编程实践能力。

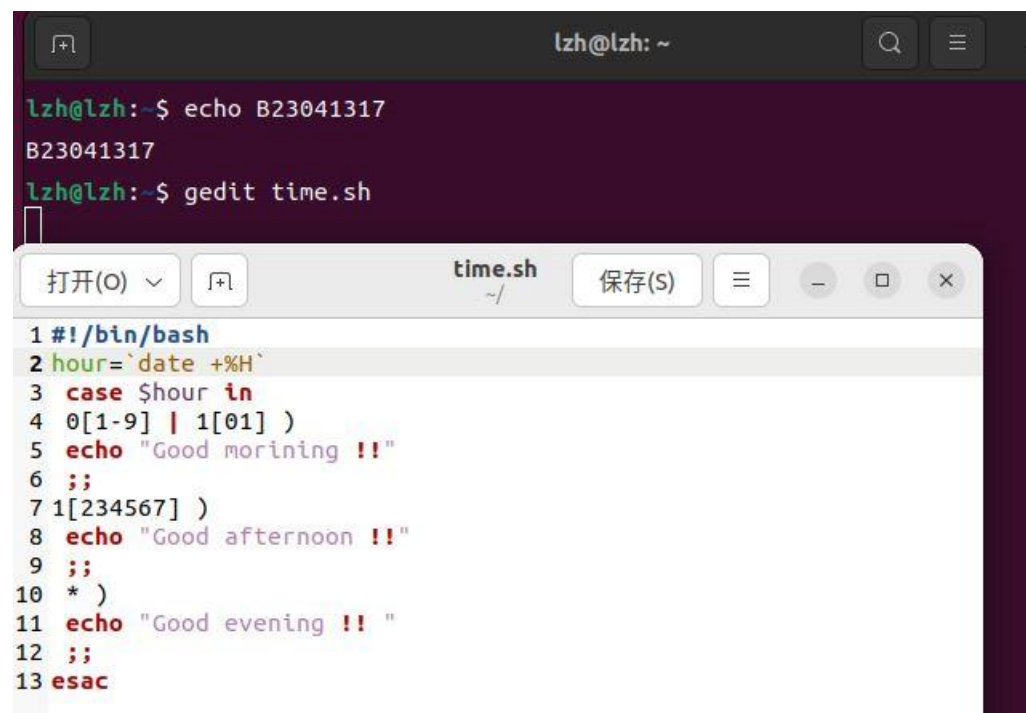
二、实验要求

- 1、编写 Shell 脚本，获取系统时间并判断当前时段（上午、下午或晚上）。
- 2、编写 Shell 脚本，接收用户输入的两个整数，比较两者大小并输出比较结果。
- 3、编写 Shell 脚本，在给定的数字列表中查找最小值并输出。
- 4、编写 Shell 脚本，统计当前目录下可执行文件的数量并输出。
- 5、编写 Shell 脚本，通过自定义函数判断一个给定的数字是否为质数，要求调用该函数并输出判断结果。

三、实验内容

(一) 获取系统时间并判断时段

- 1、通过脚本获取系统当前小时，根据小时范围判断所属时段并输出对应时间问候语
- 2、执行步骤：用 gedit 命令创建脚本文件，通过 chmod 命令赋予脚本执行权限，运行查看结果
- 3、操作截图：



The screenshot shows a terminal window and a text editor window. The terminal window displays the following commands and output:

```
lzh@lzh:~$ echo B23041317
B23041317
lzh@lzh:~$ gedit time.sh
```

The text editor window, titled "time.sh", shows the following code:

```
1 #!/bin/bash
2 hour=`date +%H`
3 case $hour in
4 0[1-9] | 1[01] )
5 echo "Good morining !!"
6 ;;
7 1[234567] )
8 echo "Good afternoon !!"
9 ;;
10 * )
11 echo "Good evening !! "
12 ;;
13 esac
```

上述图片为创建脚本文件并贴上修改后的代码

```
lzh@lzh: ~  
lzh@lzh:~$ echo B23041317  
B23041317  
lzh@lzh:~$ gedit time.sh  
lzh@lzh:~$ chmod +x time.sh  
lzh@lzh:~$ ./time.sh  
Good morning !!  
lzh@lzh:~$
```

（二）输入两个整数比较大小

- 1、目的：编写脚本实现接收用户输入的整数并比较大小
- 2、执行步骤：用 gedit 创建脚本，贴上代码后赋予可执行权限，执行脚本
- 3、操作截图：

```
lzh@lzh:~$ gedit number.sh  
1 #!/bin/sh  
2 echo "Enter the first integer:"  
3 read first  
4 echo "Enter the second integer:"  
5 read second  
6 if [ "$first" -gt "$second" ]  
7 then  
8 echo "$first is greater than $second"  
9 elif [ "$first" -lt "$second" ]  
10 then  
11 echo "$first is less than $second"  
12 else  
13 echo "$first is equal to $second"  
14 fi
```

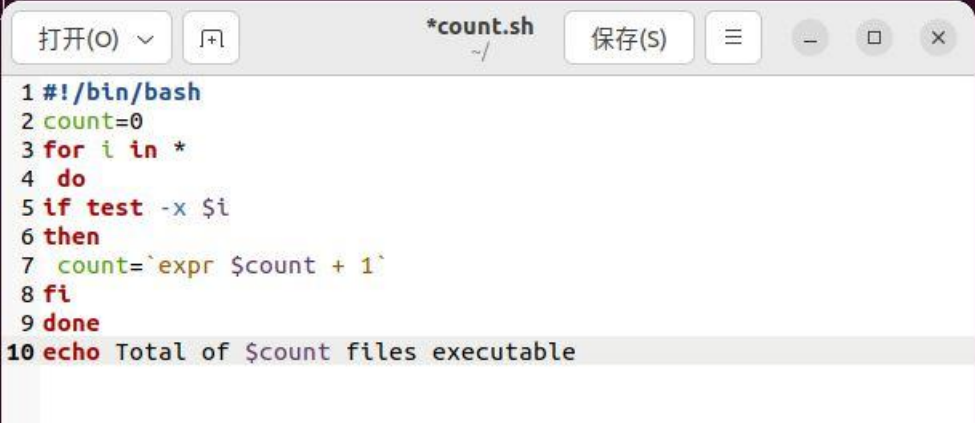


```
lzh@lzh: ~  
lzh@lzh:~$ echo B23041317  
B23041317  
lzh@lzh:~$ gedit min.sh  
lzh@lzh:~$ chmod +x min.sh  
lzh@lzh:~$ ./min.sh  
-3  
lzh@lzh:~$
```

（四）统计当前目录下可执行文件数量

- 1、目的：编写脚本遍历当前目录下所有的非隐藏文件，判断文件是否具有可执行权限，统计可执行文件总数并输出
- 2、执行步骤：用 gedit 创建脚本，贴上代码后赋予可执行权限，执行脚本
- 3、操作截图：

```
lzh@lzh:~$ echo B23041317  
B23041317  
lzh@lzh:~$ gedit count.sh
```

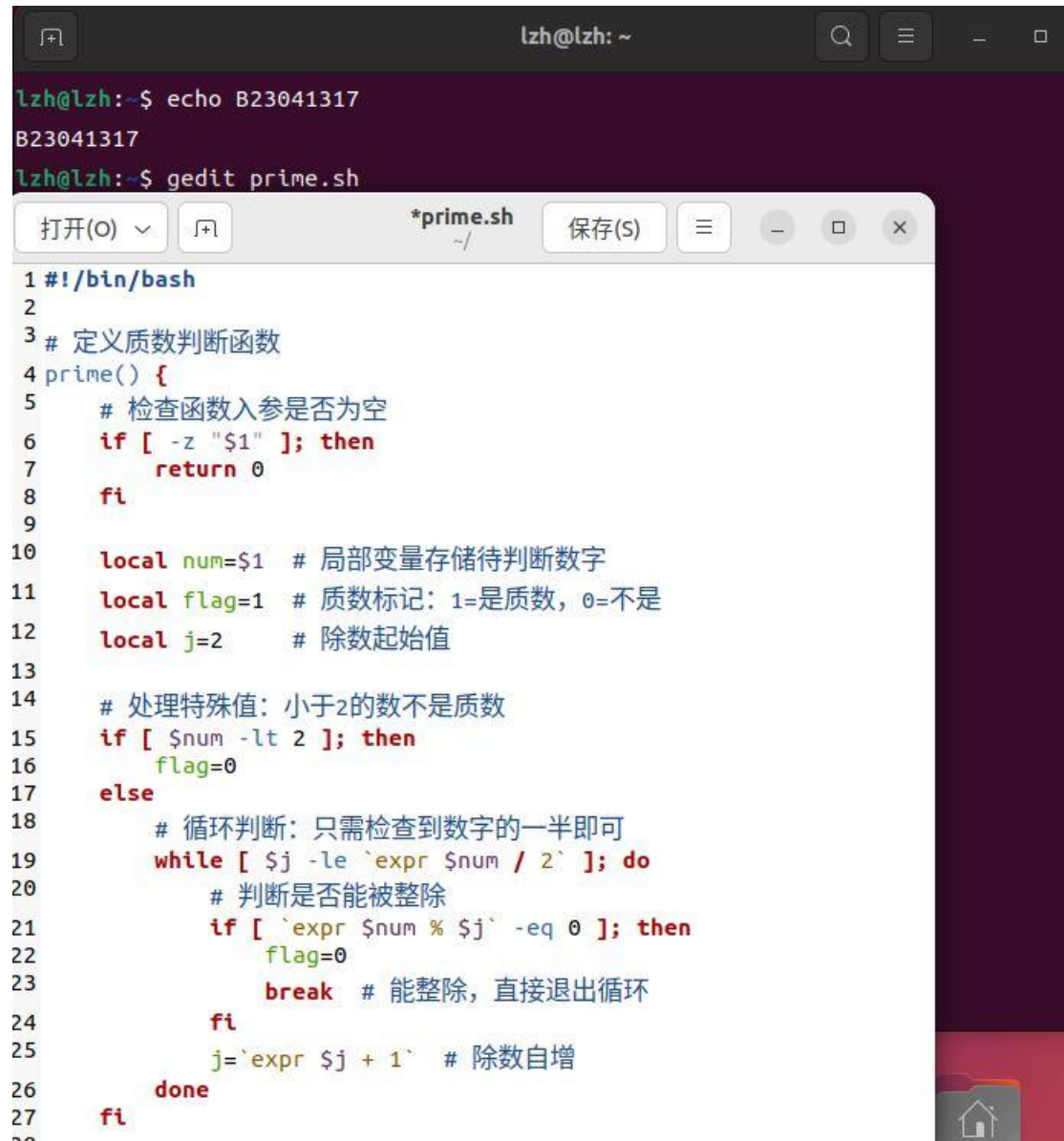


```
1 #!/bin/bash  
2 count=0  
3 for i in *  
4 do  
5 if test -x $i  
6 then  
7 count=`expr $count + 1`  
8 fi  
9 done  
10 echo Total of $count files executable
```

```
lzh@lzh:~$ echo B23041317  
B23041317  
lzh@lzh:~$ gedit count.sh  
lzh@lzh:~$ chmod +x count.sh  
lzh@lzh:~$ ./count.sh  
Total of 26 files executable  
lzh@lzh:~$
```

（五）自定义函数判断给定数字是否为质数

- 1、目的：用脚本实现判断质数的功能，脚本接受用户输入数字并判断
- 2、执行步骤：用 gedit 创建脚本，补全不完全的代码，贴上代码后赋予可执行权限，执行脚本
- 3、操作截图：



The image shows a terminal window and a gedit editor window. The terminal window displays the execution of a script that prints 'B23041317'. The gedit editor window shows the script code for a prime number checking function.

```
lzh@lzh: ~  
lzh@lzh:~$ echo B23041317  
B23041317  
lzh@lzh:~$ gedit prime.sh  
打开(O)  *prime.sh 保存(S)  
1 #!/bin/bash  
2  
3 # 定义质数判断函数  
4 prime() {  
5     # 检查函数入参是否为空  
6     if [ -z "$1" ]; then  
7         return 0  
8     fi  
9  
10    local num=$1 # 局部变量存储待判断数字  
11    local flag=1 # 质数标记: 1=是质数, 0=不是  
12    local j=2    # 除数起始值  
13  
14    # 处理特殊值: 小于2的数不是质数  
15    if [ $num -lt 2 ]; then  
16        flag=0  
17    else  
18        # 循环判断: 只需检查到数字的一半即可  
19        while [ $j -le `expr $num / 2` ]; do  
20            # 判断是否能被整除  
21            if [ `expr $num % $j` -eq 0 ]; then  
22                flag=0  
23                break # 能整除, 直接退出循环  
24            fi  
25            j=`expr $j + 1` # 除数自增  
26        done  
27    fi  
28 }
```

```
lzh@lzh: ~  
lzh@lzh:~$ echo B23041317  
B23041317  
lzh@lzh:~$ chmod +x prime.sh  
lzh@lzh:~$ ./prime.sh  
用法: ./prime.sh <正整数>  
示例: ./prime.sh 17  
lzh@lzh:~$ ./prime.sh 20  
20 is not a prime!  
lzh@lzh:~$ ./prime.sh 13  
13 is a prime!  
lzh@lzh:~$
```