1. Obtain the system time, and check whether it is in the morning, afternoon, or evening.

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

根据当前时间自动判断是早上、下午还是晚上，并输出相应的问候语。例如，如果在早上 7 点运行这个脚本，它将输出“Good morning !!”。如果在下午 3 点运行，它将输出“Good afternoon !!”。如果是在晚上 8 点运行，它将输出“Good evening !!”

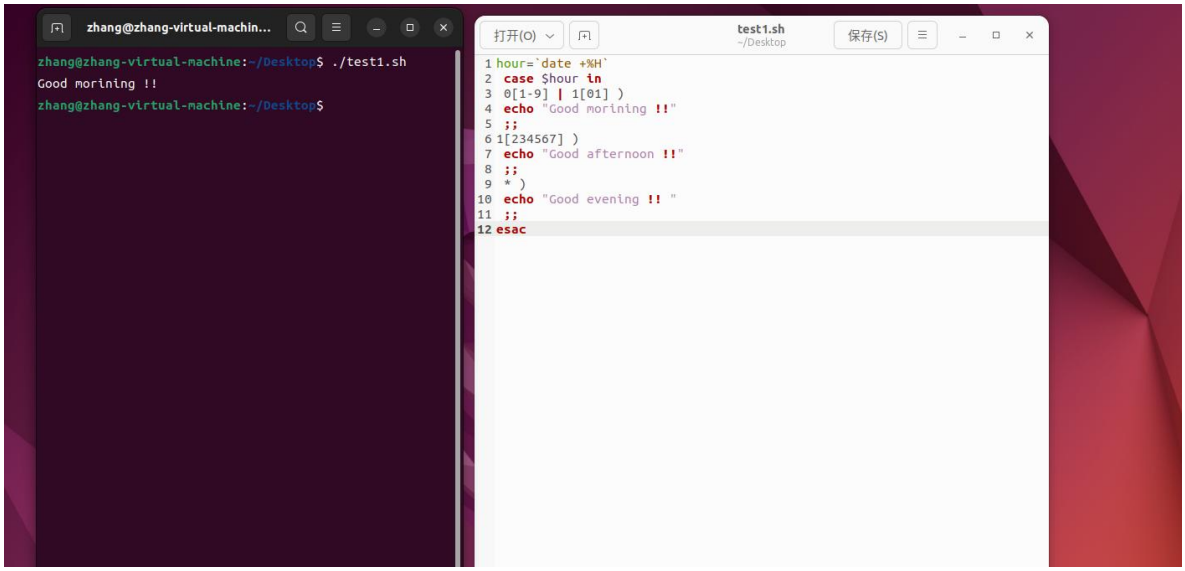

图 1 结果展示（1）

2. Input two number, check which one is greater, and output the result.

```
#!/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
```

让用户输入两个整数，然后比较它们的大小，并输出比较结果。如果第一个整数大于第二个，输出第一个整数大于第二个；如果第一个整数小于第二个，输出第一个整数小于第二个；如果两者相等，则输出两个整数相等的消息。需要注意的是，脚本中存在变量大小写的不一致问题，应该统一使用小写或大写以避免潜在的错误。

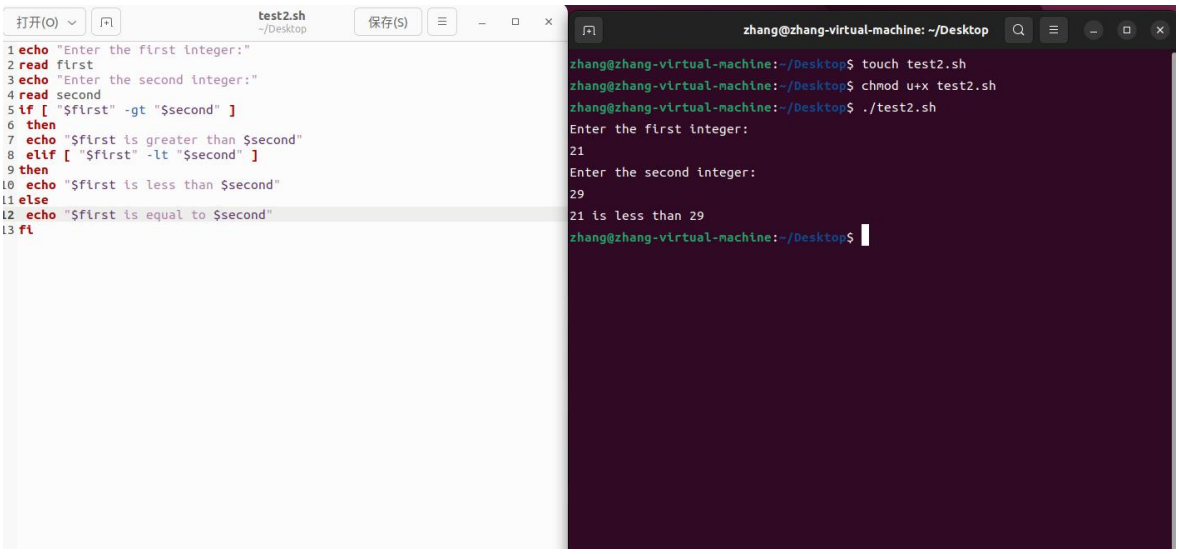

图 2 结果展示（2）

3. Find the minimal value in a given list.

```
#!/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
```

通过遍历给定的整数列表，并不断更新 smallest 变量来找出并打印出列表中的最小值。在这个例子中，最终会输出-3，因为-3 是给定整数列表中的最小值。

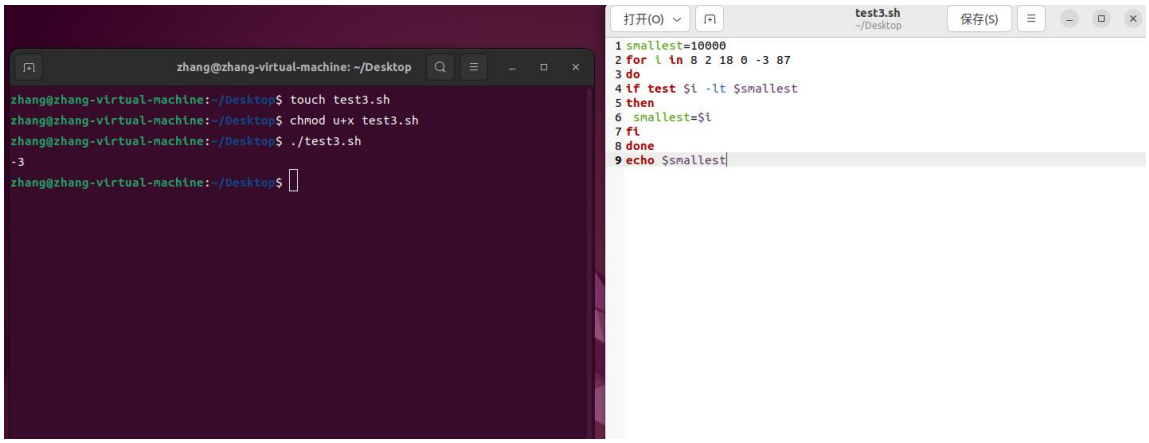

图 3 结果展示（3）

4. Calculate the number of executive file in the current directory.

```
#!/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
```

通过遍历当前目录下的所有文件，并检查每个文件是否具有可执行权限，来统计可执行文件的数量，并最终输出这个数量。这个脚本可以帮助用户快速了解当前目录下有多少文件是可以直接执行的。

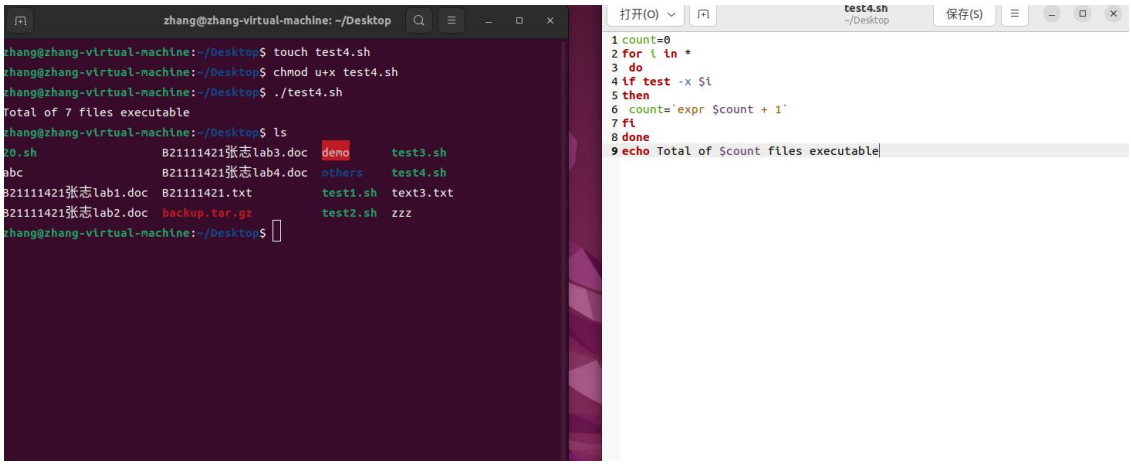

图 4 结果展示（4）

5. Check whether a given number is a prime, you have to write a function, and call the function.

```
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
```

定义一个 prime 函数来检查传入的参数是否为质数，并根据结果输出相应的消息。如果参数是质数，输出该数字是质数；如果不是，输出该数字不是质数。

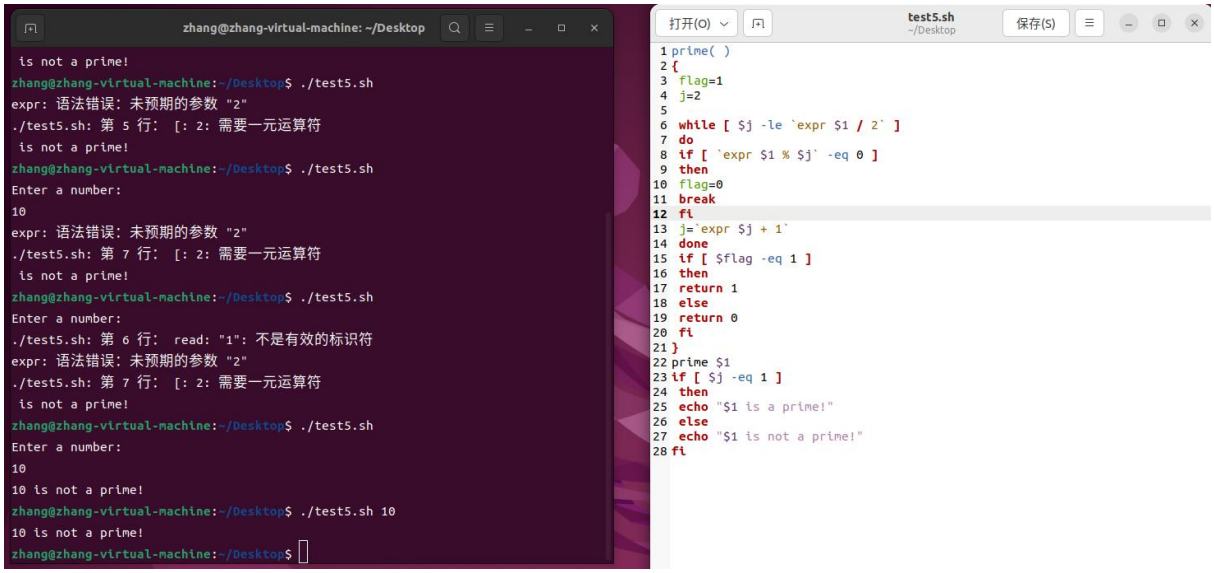
zhang@zhang-virtual-machine: ~/Desktop
is not a prime!
zhang@zhang-virtual-machine:~/Desktop$ ./test5.sh
expr: 语法错误：未预期的参数 "2"
./test5.sh: 第 5 行： [: 2: 需要一元运算符
is not a prime!
zhang@zhang-virtual-machine:~/Desktop$ ./test5.sh
Enter a number:
10
expr: 语法错误：未预期的参数 "2"
./test5.sh: 第 7 行： [: 2: 需要一元运算符
is not a prime!
zhang@zhang-virtual-machine:~/Desktop$ ./test5.sh
Enter a number:
./test5.sh: 第 6 行： read: "1": 不是有效的标识符
expr: 语法错误：未预期的参数 "2"
./test5.sh: 第 7 行： [: 2: 需要一元运算符
is not a prime!
zhang@zhang-virtual-machine:~/Desktop$ ./test5.sh
Enter a number:
10
10 is not a prime!
zhang@zhang-virtual-machine:~/Desktop$ ./test5.sh 10
10 is not a prime!
zhang@zhang-virtual-machine:~/Desktop$
打开(O)
test5.sh
~/Desktop
保存(S)
1 prime( )
2 {
3 flag=1
4 j=2
5
6 while [$j -le `expr $1 / 2`]
7 do
8 if [`expr $1 % $j` -eq 0]
9 then
10 flag=0
11 break
12 fi
13 j=`expr $j + 1`
14 done
15 if [$flag -eq 1]
16 then
17 return 1
18 else
19 return 0
20 fi
21 }
22 prime $1
23 if [$j -eq 1]
24 then
25 echo "$1 is a prime!"
26 else
27 echo "$1 is not a prime!"
28 fi

图 5 结果展示（5）