

# Experiment 2

B22041026 刘宇凡

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

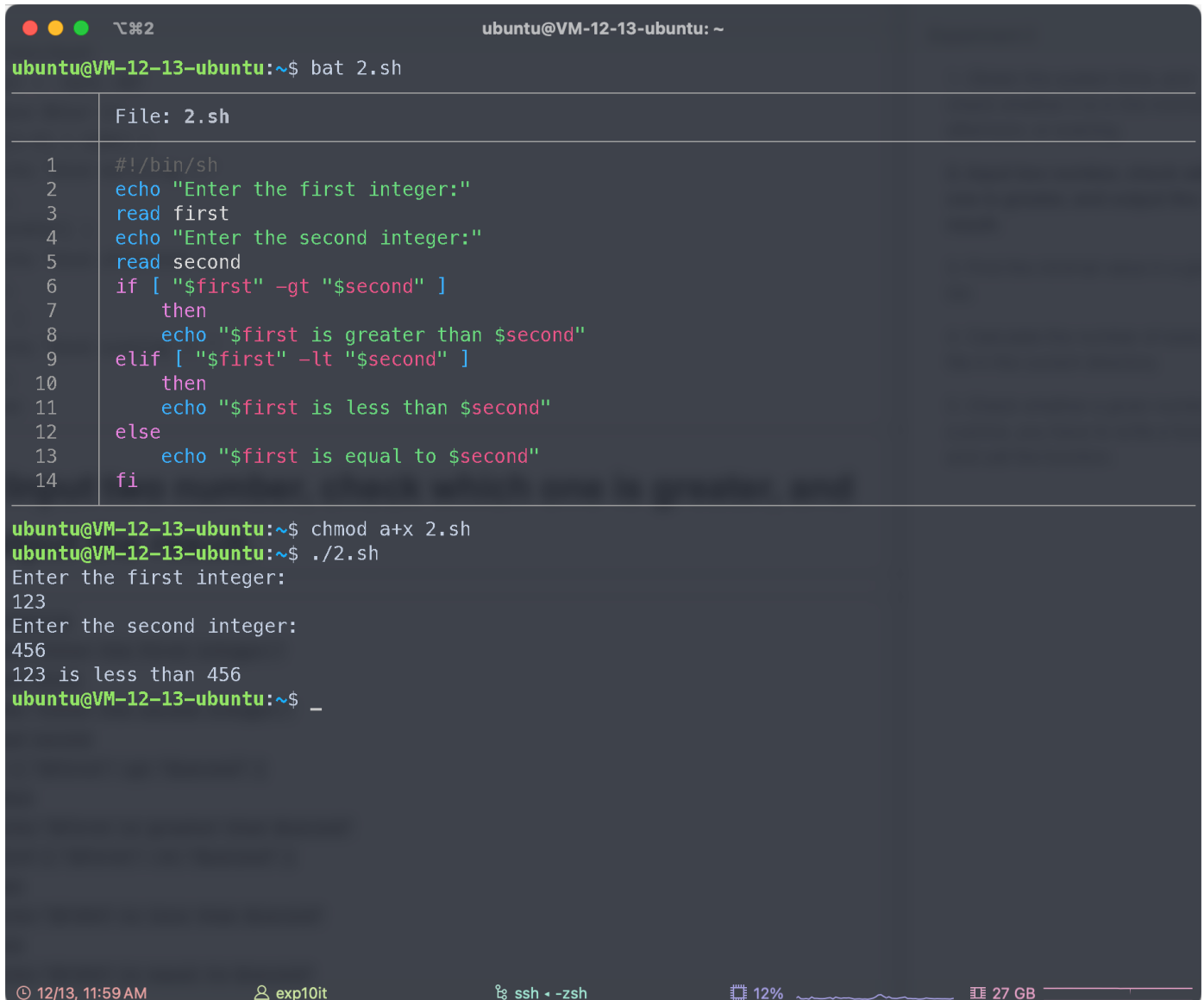
```
#!/bin/bash
hour=`date +%H`
case $hour in
0[1-9] | 1[01] )
    echo "Good morining !!"
;;
1[234567] )
    echo "Good afternoon !!"
;;
* )
    echo "Good evening !! "
;;
esac
```

```
ubuntu@VM-12-13-ubuntu: ~  
ubuntu@VM-12-13-ubuntu:~$ bat 1.sh  
File: 1.sh  
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  
  
ubuntu@VM-12-13-ubuntu:~$ chmod a+x 1.sh  
ubuntu@VM-12-13-ubuntu:~$ ./1.sh  
Good morining !!  
ubuntu@VM-12-13-ubuntu:~$ _
```

**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
```

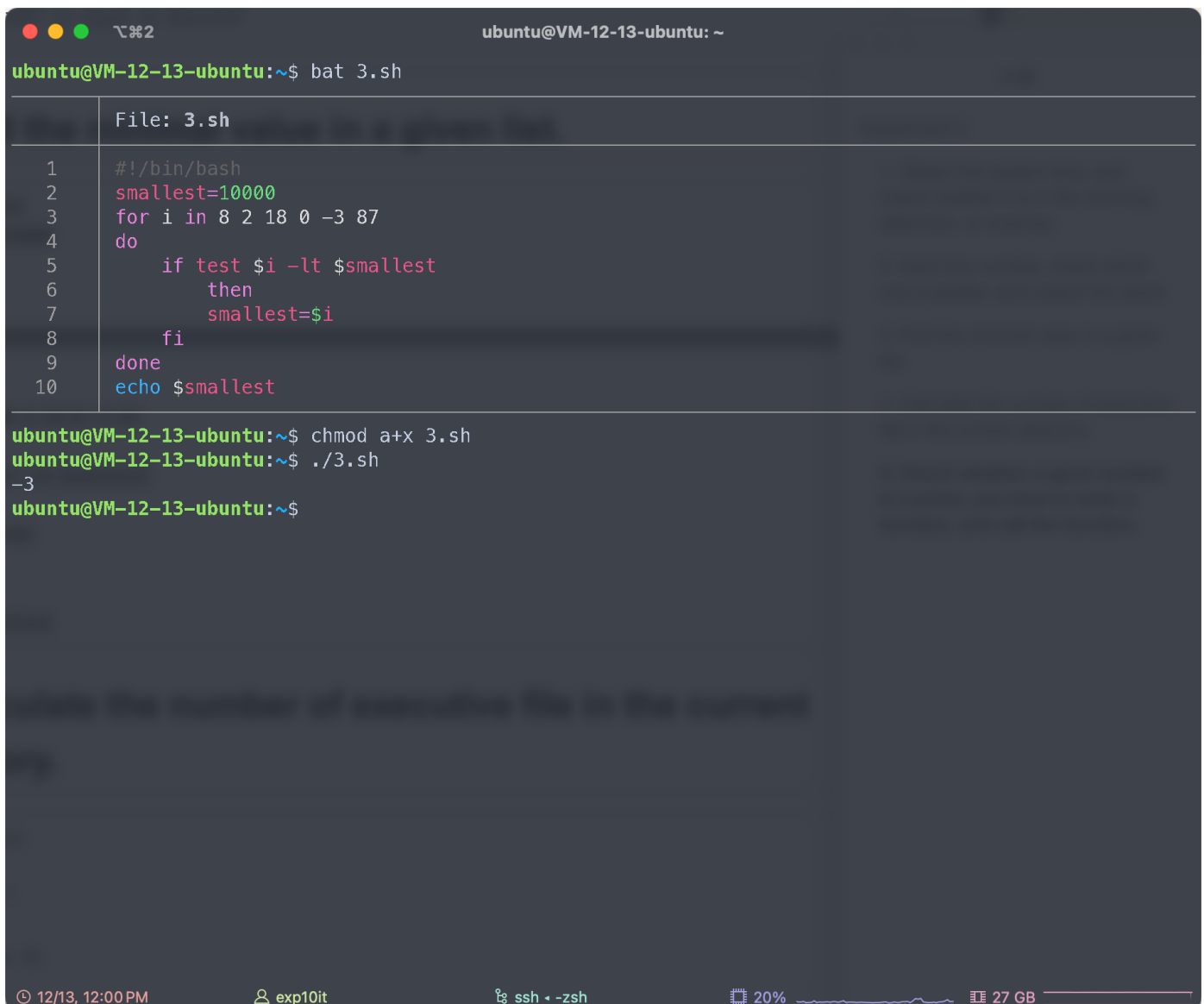


```
ubuntu@VM-12-13-ubuntu: ~  
ubuntu@VM-12-13-ubuntu:~$ bat 2.sh  
File: 2.sh  
1  #!/bin/sh  
2  echo "Enter the first integer:"  
3  read first  
4  echo "Enter the second integer:"  
5  read second  
6  if [ "$first" -gt "$second" ]  
7  then  
8      echo "$first is greater than $second"  
9  elif [ "$first" -lt "$second" ]  
10 then  
11     echo "$first is less than $second"  
12 else  
13     echo "$first is equal to $second"  
14 fi  
ubuntu@VM-12-13-ubuntu:~$ chmod a+x 2.sh  
ubuntu@VM-12-13-ubuntu:~$ ./2.sh  
Enter the first integer:  
123  
Enter the second integer:  
456  
123 is less than 456  
ubuntu@VM-12-13-ubuntu:~$ _
```

© 12/13, 11:59 AM    exp10it    ssh • -zsh    12%    27 GB

### 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
```

A terminal window titled 'ubuntu@VM-12-13-ubuntu: ~' with a dark background. The user runs 'bat 3.sh', which displays the script content with line numbers 1 through 10. Then, the user runs 'chmod a+x 3.sh' and './3.sh', which outputs '-3'. The terminal status bar at the bottom shows '12/13, 12:00 PM', 'exp10it', 'ssh - zsh', '20%' CPU usage, and '27 GB' memory usage.

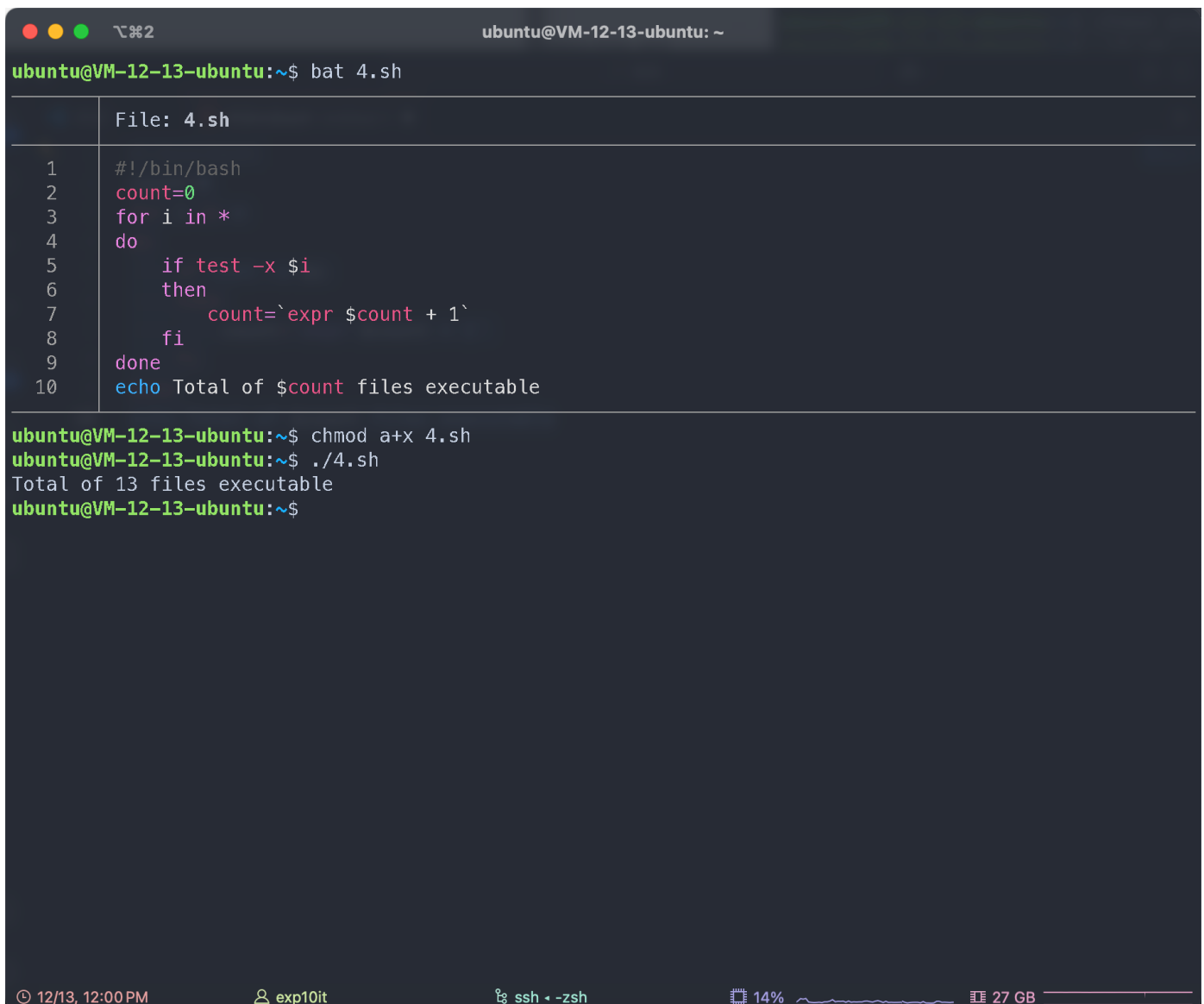
```
ubuntu@VM-12-13-ubuntu:~$ bat 3.sh
File: 3.sh
1  #!/bin/bash
2  smallest=10000
3  for i in 8 2 18 0 -3 87
4  do
5      if test $i -lt $smallest
6      then
7          smallest=$i
8      fi
9  done
10 echo $smallest

ubuntu@VM-12-13-ubuntu:~$ chmod a+x 3.sh
ubuntu@VM-12-13-ubuntu:~$ ./3.sh
-3
ubuntu@VM-12-13-ubuntu:~$
```

12/13, 12:00 PM    exp10it    ssh - zsh    20%    27 GB

**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
```



The screenshot shows a terminal window with the following content:

```
ubuntu@VM-12-13-ubuntu: ~$ bat 4.sh
File: 4.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

ubuntu@VM-12-13-ubuntu:~$ chmod a+x 4.sh
ubuntu@VM-12-13-ubuntu:~$ ./4.sh
Total of 13 files executable
ubuntu@VM-12-13-ubuntu:~$
```

The terminal window has a title bar with window control buttons and the text "ubuntu@VM-12-13-ubuntu: ~". The status bar at the bottom shows the date and time "12/13, 12:00 PM", the username "exp10it", the connection type "ssh - zsh", the CPU usage "14%", and the memory usage "27 GB".

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

```

prime() {
    if [ -z "$1" ]; then
        echo "Error: No input provided."
        return 0
    fi

    flag=1
    j=2

    while [ $j -le $(( $1 / 2 )) ]; do
        if [ $(( $1 % $j )) -eq 0 ]; then
            flag=0
            break
        fi
        j=$(( $j + 1 ))
    done

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

if [ -z "$1" ]; then
    echo "Usage: $0 <number>"
    exit 1
fi

prime $1

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

```

```
ubuntu@VM-12-13-ubuntu: ~  
ubuntu@VM-12-13-ubuntu:~$ bash 5.sh  
Usage: 5.sh <number>  
ubuntu@VM-12-13-ubuntu:~$ bash 5.sh 123  
123 is not a prime!  
ubuntu@VM-12-13-ubuntu:~$ bash 5.sh 124  
124 is not a prime!  
ubuntu@VM-12-13-ubuntu:~$ bash 5.sh 125  
125 is not a prime!  
ubuntu@VM-12-13-ubuntu:~$ bash 5.sh 126  
126 is not a prime!  
ubuntu@VM-12-13-ubuntu:~$ bash 5.sh 127  
127 is a prime!  
ubuntu@VM-12-13-ubuntu:~$ _
```

12/13, 12:08 PM    exp10it    ssh - zsh    17%    28 GB