

Experiment 2

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

1. 在终端中输入命令: nano time_check.sh, 按下回车后, 进入 nano 编辑界面。将文本输入进去后保存脚本并退出 nano 编辑器。

2. 授权命令: 在终端输入 chmod u+x time_check.sh, 按下回车。

3. 运行命令: 在终端输入./time_check.sh

结果: 由于当时时间为 19: 25, 所以终端显示: Good evening !!

The screenshot shows a terminal window with two parts. The top part is the nano text editor displaying a shell script named 'time_check.sh'. The script uses the date command to get the hour and then prints 'Good morning !!!', 'Good afternoon !!!', or 'Good evening !!' based on the hour. The bottom part is a terminal session where the user runs the script, changes its permissions, and executes it, resulting in the output 'Good evening !!'.

```
chenjie@chenjie-VMware-Virtual-Platform:~/桌面$ nano time_check.sh
GNU nano 7.2                               time_check.sh
#!/bin/bash
hour=`date +%H`
case $hour in
0[1-9] | 1[01] )
echo "Good morning !!!"
;;
1[234567] )
echo "Good afternoon !!!"
;;
* )
echo "Good evening !! "
;;
esac

[ 已写入 13 行 ]
^G 帮助      ^O 写入      ^W 搜索      ^K 剪切      ^T 执行命令      ^C 位置
^X 离开      ^R 读档      ^\ 替换      ^U 粘贴      ^J 对齐      ^/ 跳行

chenjie@chenjie-VMware-Virtual-Platform:~/桌面$ nano time_check.sh
chenjie@chenjie-VMware-Virtual-Platform:~/桌面$ chmod u+x time_check.sh
chenjie@chenjie-VMware-Virtual-Platform:~/桌面$ ./time_check.sh
Good evening !!
chenjie@chenjie-VMware-Virtual-Platform:~/桌面$
```

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

1. 在终端输入命令，创建名为 number_compare.sh 的脚本，将代码输入进去保存并退出。
2. 授权命令：在终端输入 chmod u+x time_check.sh，按下回车。
3. 运行命令：在终端输入 ./number_compare.sh
4. 程序开始运行，此时输入两个随机的数，如 2 和 9，然后回车。
5. 结果：2 小于 9



```

chenjie@chenjie-VMware-Virtual-Platform:~/桌面
GNU nano 7.2          number_compare.sh
#!/bin/sh
echo "请输入第一个整数: "
read first
echo "请输入第二个整数: "
read second

if [ "$first" -gt "$second" ]
then
echo "$first 大于 $second"
elif [ "$first" -lt "$second" ]
then
echo "$first 小于 $second"
else
echo "$first 等于 $second"
fi

[ 已写入 15 行 ]
^G 帮助      ^O 写入      ^W 搜索      ^K 剪切      ^T 执行命令      ^C 位置
^X 离开      ^R 读档      ^\ 替换      ^U 粘贴      ^J 对齐      ^/ 跳行

```

```

chenjie@chenjie-VMware-Virtual-Platform:~/桌面$ nano number_compare.sh
chenjie@chenjie-VMware-Virtual-Platform:~/桌面$ chmod u+x number_compare.sh
chenjie@chenjie-VMware-Virtual-Platform:~/桌面$ ./number_compare.sh
请输入第一个整数:
2
请输入第二个整数:
9
2 小于 9
chenjie@chenjie-VMware-Virtual-Platform:~/桌面$ 

```

(3). Find the minimal value in a given list.

1. 终端输入命令，创建 find_min.sh 脚本
2. 授权命令：在终端输入 chmod u+x time_check.sh，按下回车。
3. 运行命令：在终端输入 ./find_min.sh。
4. 结果：终端输出：列表 8 2 18 0 -3 87 中的最小值是：-3

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "chenjie@chenjie-VMware-Virtual-Platform: ~/桌面". Inside the terminal, the user has run the command "nano find_min.sh" to edit a shell script. The script content is as follows:

```
GNU nano 7.2                               find_min.sh
#!/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 "列表 8 2 18 0 -3 87 中的最小值是: $smallest"
```

Below the terminal window, there is a status bar with various keyboard shortcuts. The terminal window itself shows the command "nano find_min.sh" being run, followed by "chmod u+x find_min.sh", and finally "./find_min.sh". The output of the script execution is displayed as "列表 8 2 18 0 -3 87 中的最小值是: -3".

(4). Calculate the number of executive file in the currentirectory.

- 1.终端输入命令，创建 count_exec.sh 脚本。
- 2.授权命令：在终端输入 chmod u+x time_check.sh，按下回车。
- 3.运行命令：在终端输入./count_exec.sh。
- 4.结果：终端输出：当前目录（家目录）下的可执行文件总数为：13 个

The screenshot shows a terminal window titled "chenjie@chenjie-VMware-Virtual-Platform: ~/桌面". The window contains a nano editor session for a file named "count_exec.sh". The script content is:

```
GNU nano 7.2                               count_exec.sh
#!/bin/bash
count=0
for i in *
do
if test -x $i
then
count=$(expr $count + 1)
fi
done
echo "当前目录（家目录）下的可执行文件总数为: $count 个"
```

The nano editor interface includes a status bar at the bottom with keyboard shortcuts for various operations like Help (^G), Write (^O), Search (^W), Cut (^K), Execute (^T), Position (^C), Exit (^X), Read (^R), Replace (^R), Paste (^U), Align (^J), and Jump (^L). The status bar also indicates "[已写入 10 行]" (10 lines written).

Below the editor, the terminal command history and output are shown:

```
列数 8 2 10 0 4 5 87 中的最小值是: ^J
chenjie@chenjie-VMware-Virtual-Platform:~/桌面$ nano count_exec.sh
chenjie@chenjie-VMware-Virtual-Platform:~/桌面$ chmod u+x count_exec.sh
chenjie@chenjie-VMware-Virtual-Platform:~/桌面$ ./count_exec.sh
当前目录（家目录）下的可执行文件总数为: 13 个
chenjie@chenjie-VMware-Virtual-Platform:~/桌面$
```

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

- 1.终端输入命令，创建 is_prime.sh 脚本。
- 2.授权命令：在终端输入 chmod u+x time_check.sh，按下回车。
- 3.运行命令：在终端输入./is_prime.sh 7。
- 4.结果：7 是素数！

```
chenjie@chenjie-Virtual-Platform: ~/桌面
```

```
GNU nano 7.2          is_prime.sh
```

```
#!/bin/bash
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

```

[已写入 32 行]

```
^G 帮助      ^O 写入      ^W 搜索      ^K 剪切      ^T 执行命令      ^C 位置
^X 离开      ^R 读档      ^\ 替换      ^U 粘贴      ^J 对齐      ^/ 跳行
```

```
chenjie@chenjie-Virtual-Platform:~/桌面$ nano is_prime.sh
chenjie@chenjie-Virtual-Platform:~/桌面$ chmod u+x is_prime.sh
chenjie@chenjie-Virtual-Platform:~/桌面$ ./is_prime.sh 7
7 是素数!
chenjie@chenjie-Virtual-Platform:~/桌面$
```