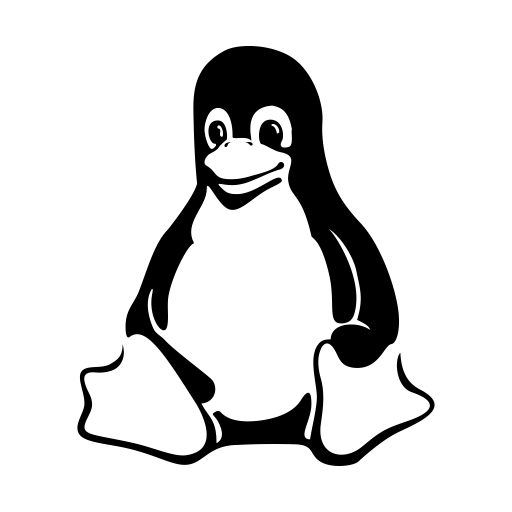
Linux高级程序设计

作业2



学号：0x091395

姓名：uknowho

年级：2014

班级：软件工程（3）班

# 思考与实验

1. 创建一个文件，其中包含了一个使用date和who命令的shell脚本，每条命令写在一个行。使得文件可执行，然后运行这个脚本。写出完成这项工作的所有步骤。

1 #!/bin/bash

2 #filename:1.sh

3 #创建一个文件，其中包含了一个使用date和who命令的shell脚本

4 #每条命令写在一个行。使得文件可执行，然后运行这个脚本。

5 echo Hello `who | grep -o 'u.\*o' | head -n 1`!

6 echo It is `date`

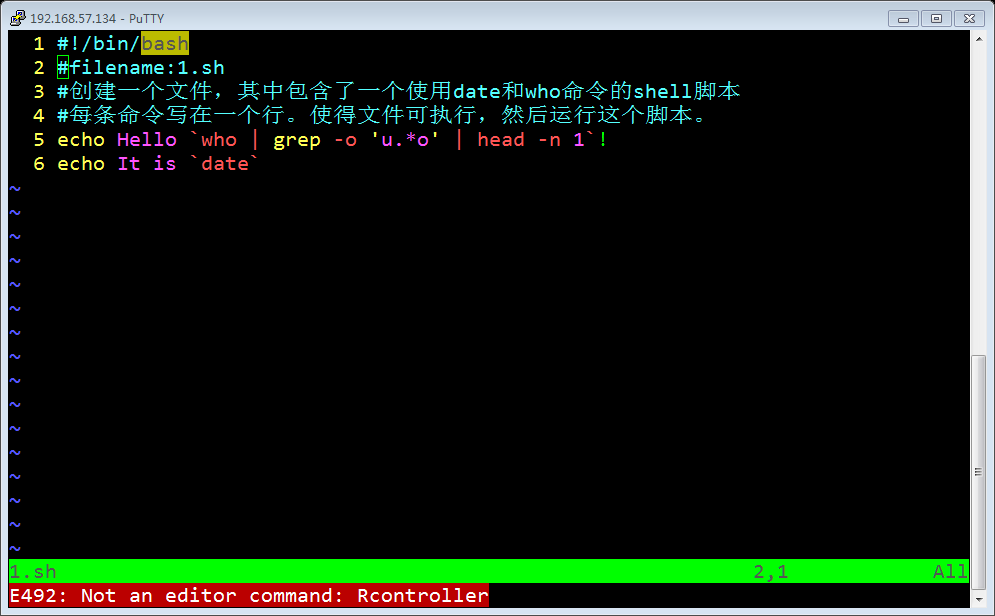
Tips：

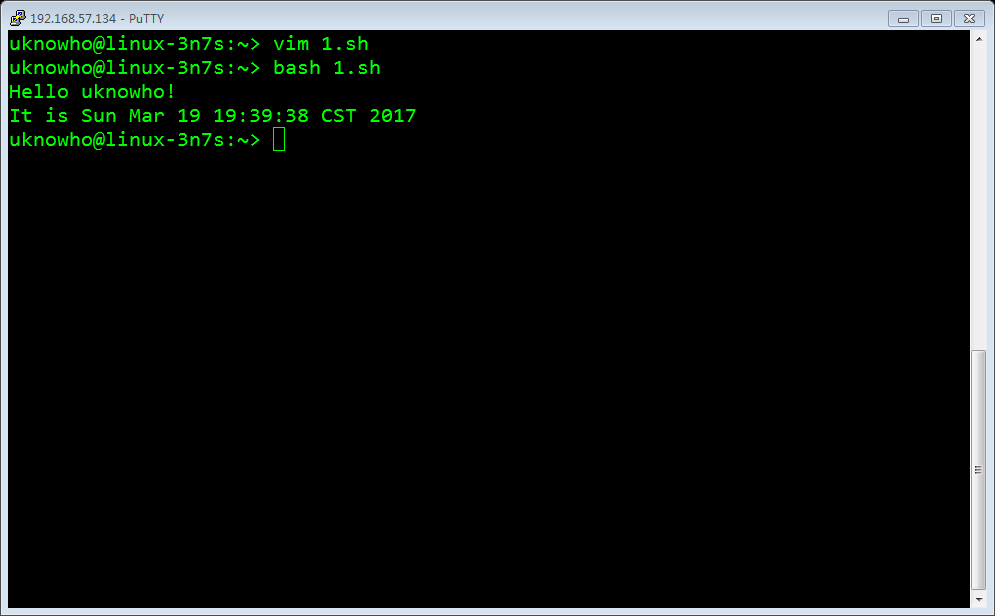
--who命令功能说明：显示目前登入系统的用户信息。

--使用反引号[` `]执行shell命令

--grep –o 只匹配输入的字符

--head –n 控制输出行数





使得1.sh变为可执行文件 输入命令 chmod a+x 1.sh 即可

1. 把echo“Hello，world”命令的输出赋值给myname变量并打印出它的值。写出完成这项工作的所有命令。

1 #!/bin/bash

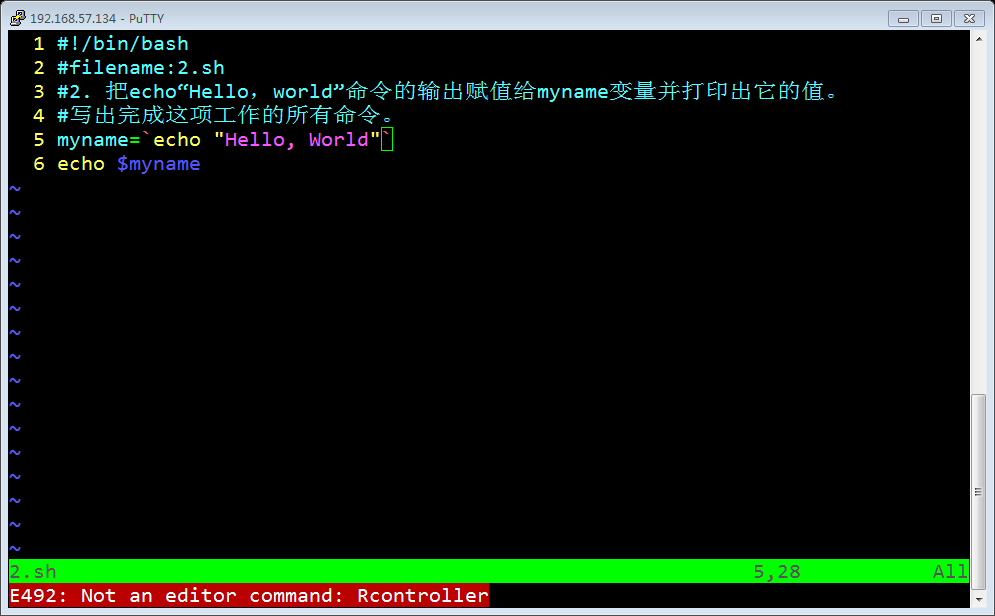
2 #filename:2.sh

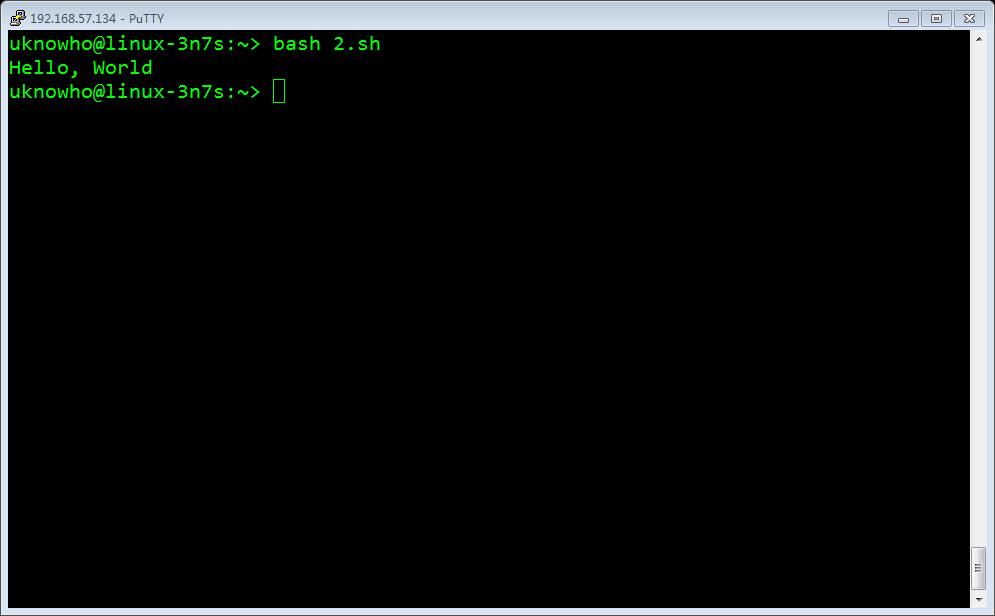
3 #2. 把echo“Hello，world”命令的输出赋值给myname变量并打印出它的值。

4 #写出完成这项工作的所有命令。

5 myname=`echo "Hello, World"`

6 echo $myname





1. 把myname变量的值复制到另一个变量anyname中，使anyname变量变为只读，对myname和anyname两个变量使用unset命令。这将有什么结果？

1 #!/bin/bash

2 #filename:3.sh

3 #3. 把myname变量的值复制到另一个变量anyname中

4 #使anyname变量变为只读，对myname和anyname两个变量使用unset命令。

5 #这将有什么结果？

6 myname=`echo "Hello, World"`

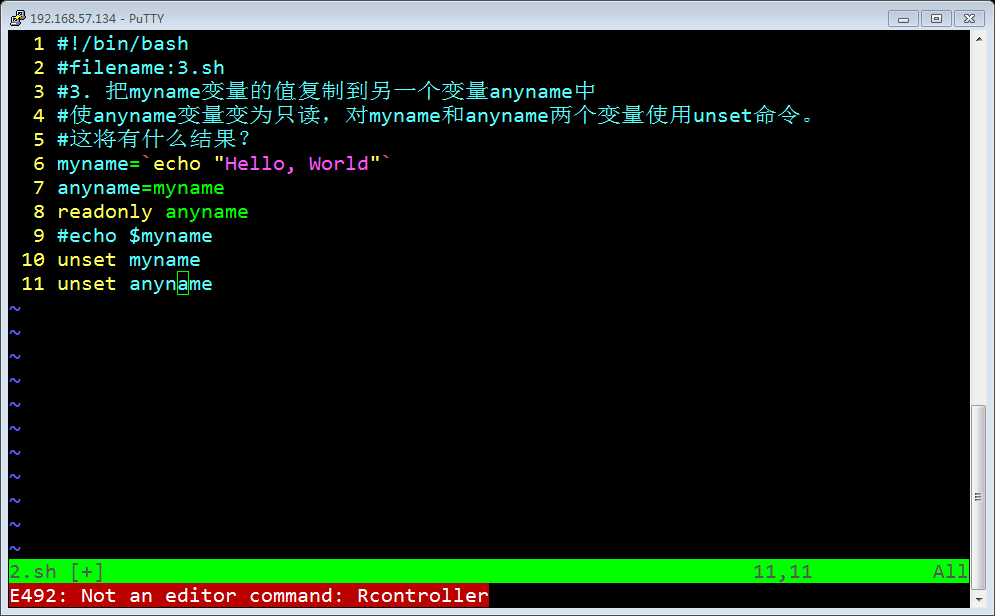
7 anyname=myname

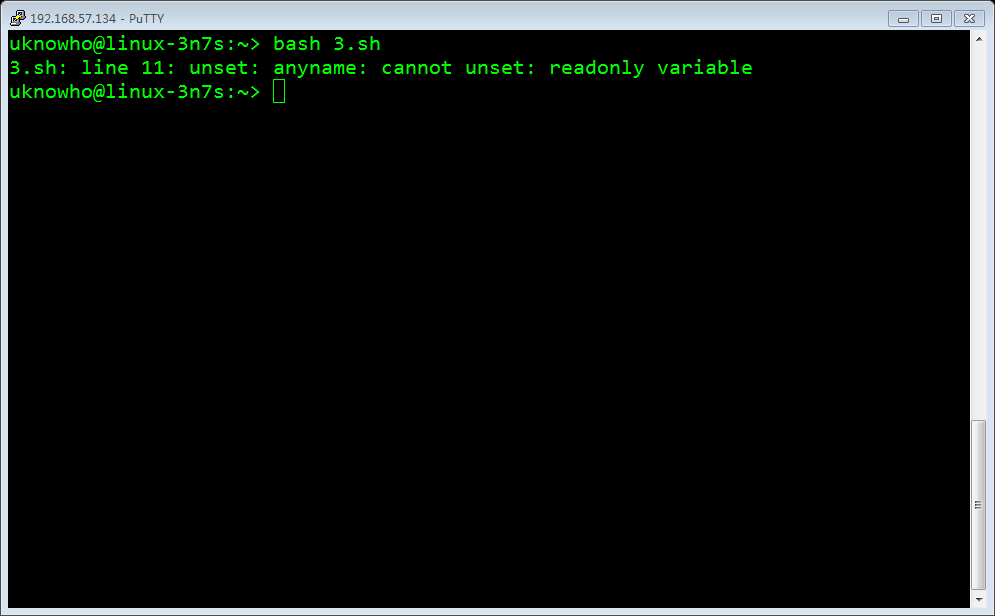
8 readonly anyname

9 #echo $myname

10 unset myname

11 unset anyname





Tip

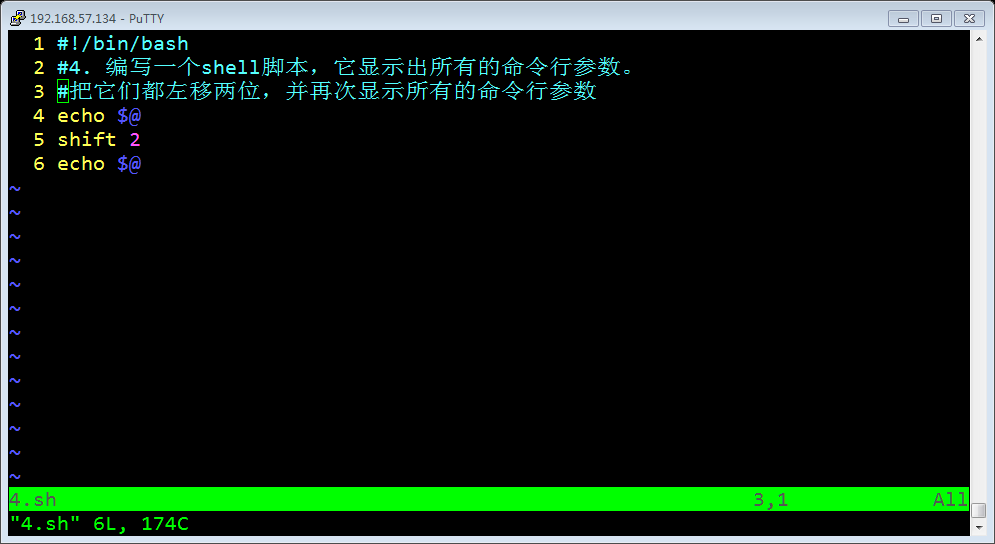
readyonly 只读变量 只能被赋值一次。

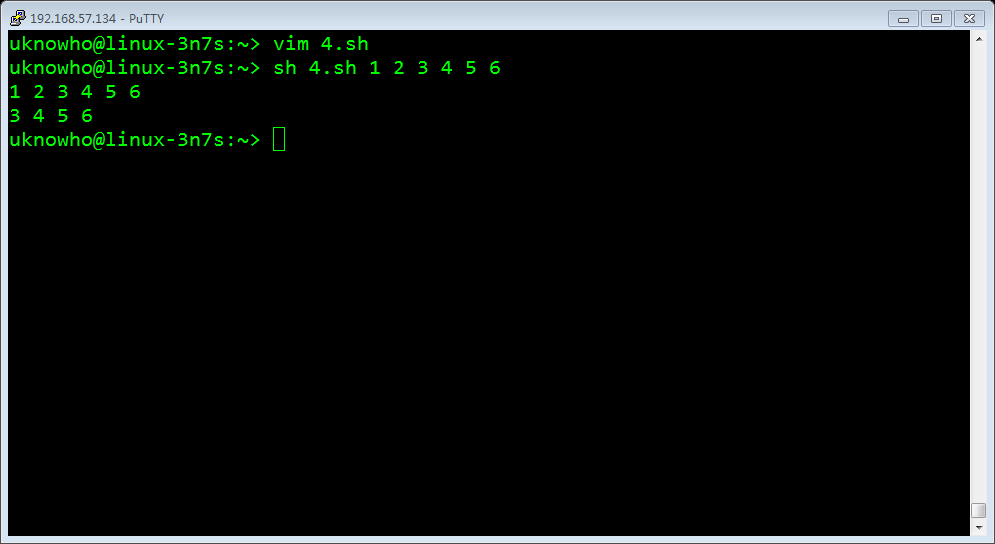
1. 编写一个shell脚本，它显示出所有的命令行参数。把它们都左移两位，并再次显示所有的命令行参数

echo $@

shitft 2

echo $@





1. 编写一个shell脚本，它带一个命令行参数，这个参数是一个文件。如果这个文件是一个普通文件，则打印文件所有者的名字和最后的修改日期。如果程序带有多个参数，则输出出错信息

if [[ $# -ne "1" || ! -f $\* ]];then

echo "请输入一个文件参数"

exit 1

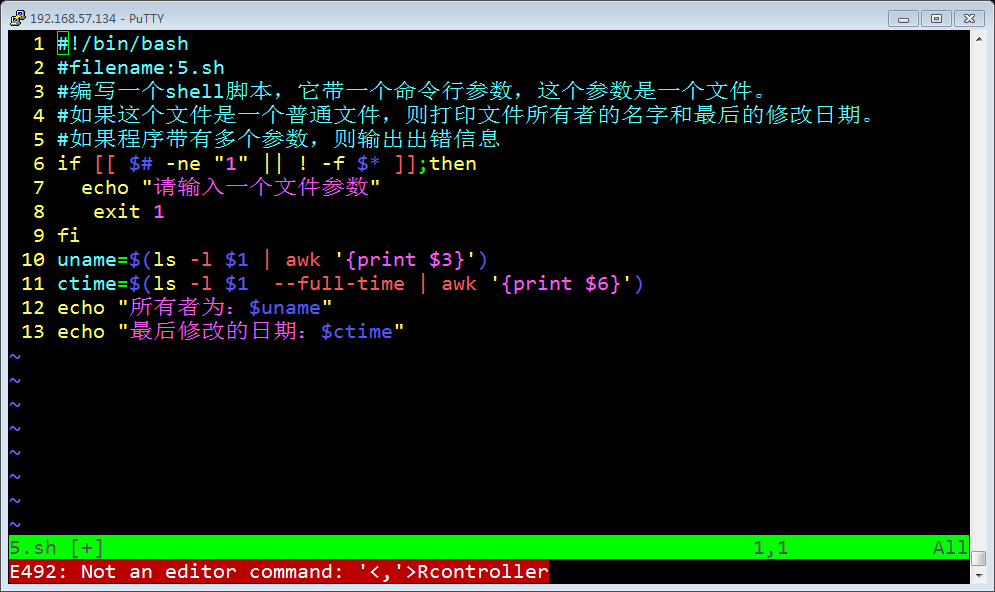
fi

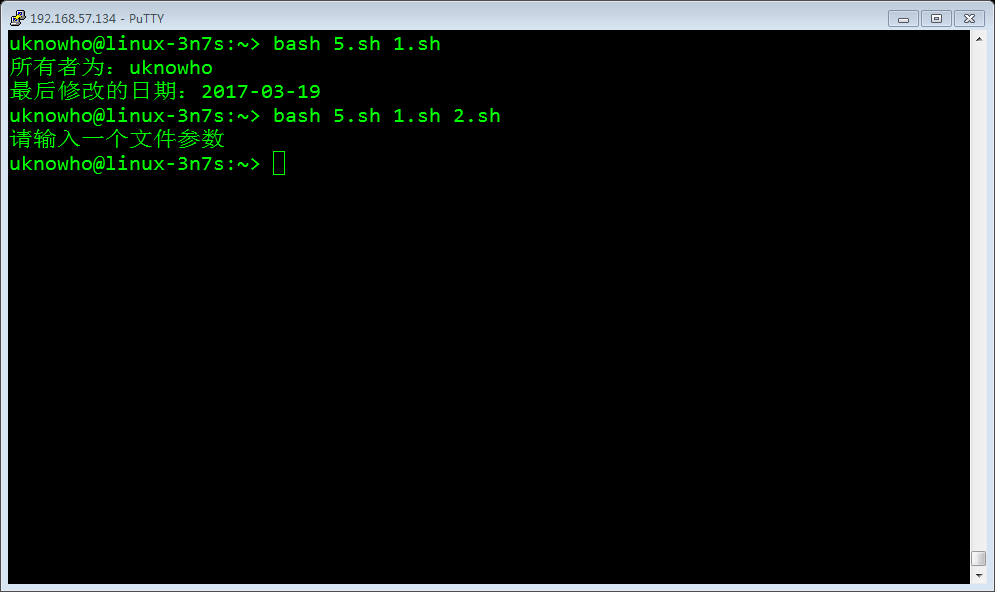
uname=$(ls -l $1 | awk '{print $3}')

ctime=$(ls -l $1 --full-time | awk '{print $6}')

echo "所有者为：$uname"

echo "最后修改的日期：$ctime"





Tips

--双中括号[[ ……. ]] 是 bash 程序语言的关键字。并不是一个命令，[[ ]] 结构比[ ]结构更加通用。在[[和]]之间所有的字符都不会发生文件名扩展或者单词分割，但是会发生参数扩展和命令替换。

使用[[ ... ]]条件判断结构，而不是[ ... ]，能够防止脚本中的许多逻辑错误。比如，&&、||、<和> 操作符能够正常存在于[[ ]]条件判断结构中，但是如果出现在[ ]结构中的话，会报错。

bash把双中括号中的表达式看作一个单独的元素，并返回一个退出状态码。

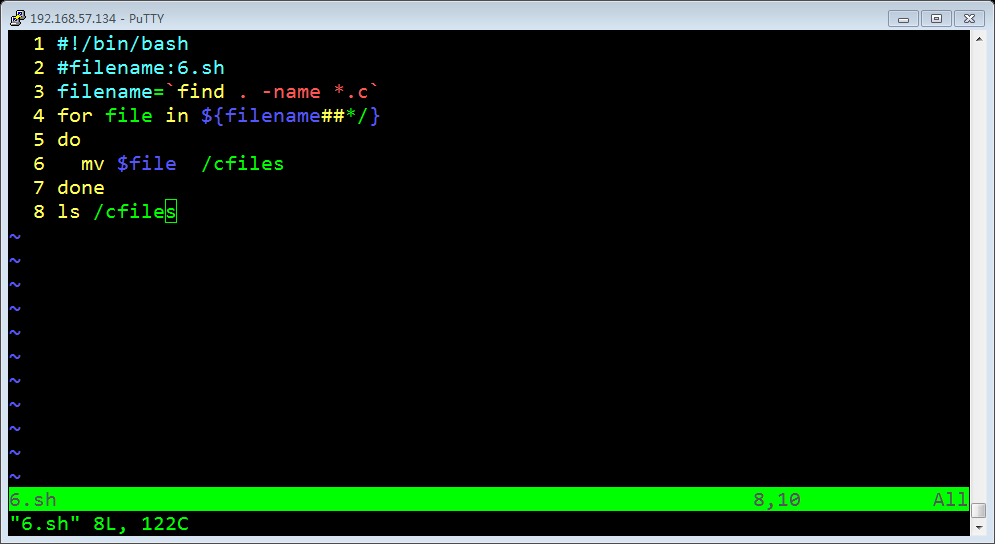
--$# 代表参数的个数

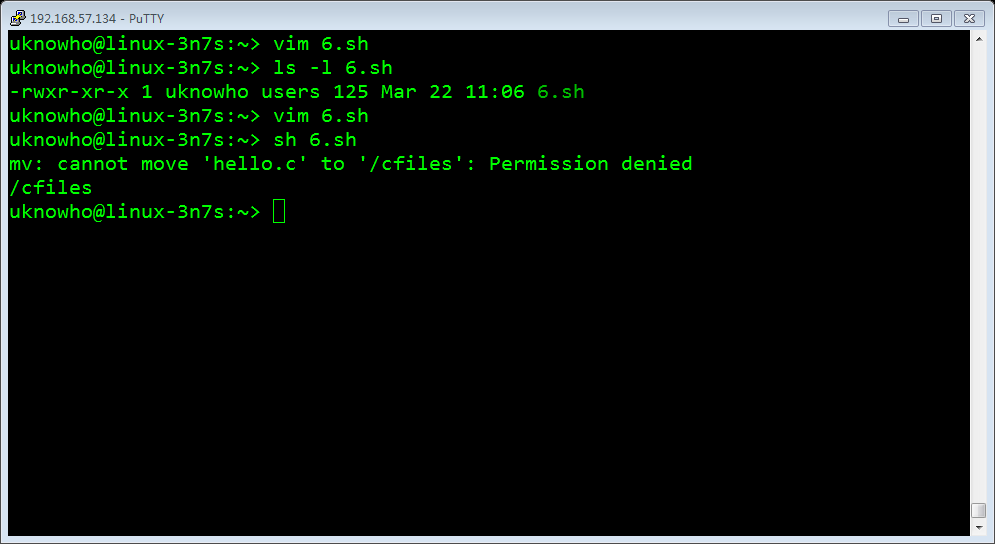
--$\* 或 $@ 代表参数列表中所有的参数

--awk ‘{print $n}’ 打印第n个字段

--ls –l $1 ????????????????????????????

1. 编写一个bash脚本程序，用for循环实现将当前目录下的所有.c文件移到指定的目录下，最后在显示器上显示指定目录下的文件和目录。



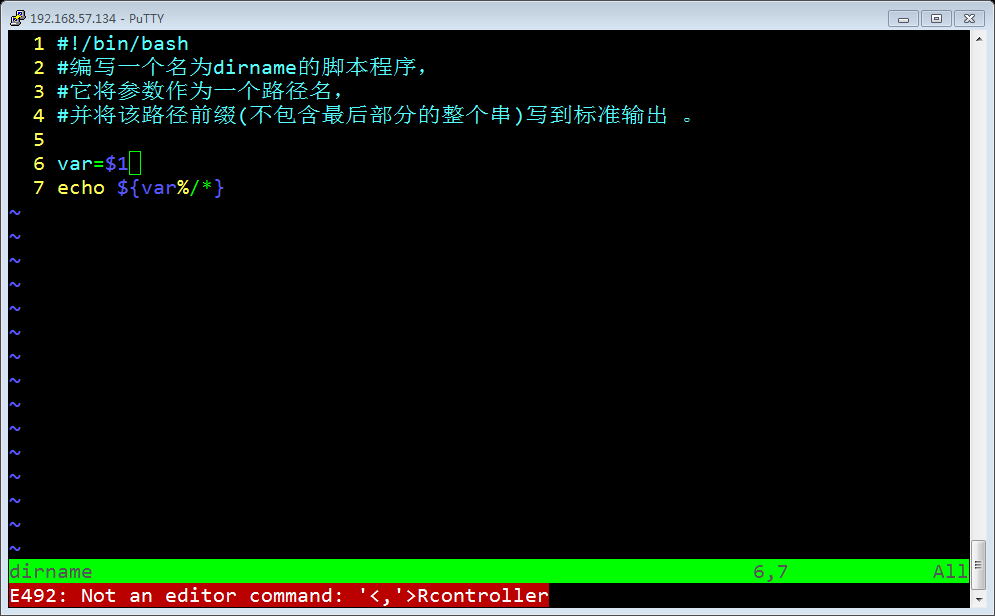


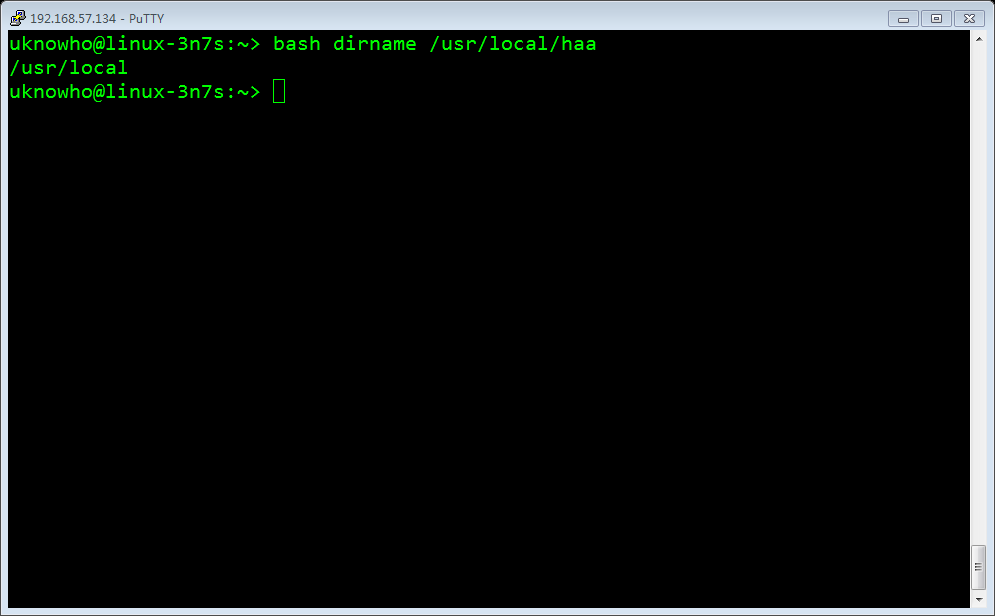
??????????????????????

1. 编写一个名为dirname的脚本程序，它将参数作为一个路径名，并将该路径前缀(不包含最后部分的整个串)写到标准输出 。

str=$1

echo ${str%/\*}





Tips

# ->左删除 \*/

% ->右删除 /\*

## %% 代表最大匹配

1. 编写一个累加器脚本程序，用Fiboracci数列的前10个数做参数。

1 #!/bin/bash

2 #编写一个累加器脚本程序，用Fiboracci数列的前10个数做参数。

3 sum=0

4 Fib=(0 1 1 2 3 5 8 13 21 34 )

5 for num in ${Fib[@]}

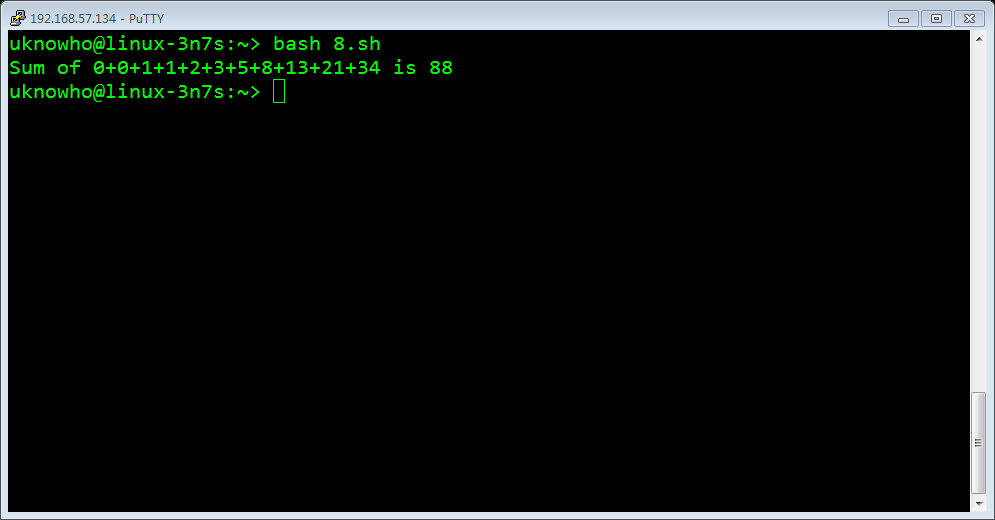
6 do

7 sum=$sum+$num

8 done

9 echo Sum of ${sum} is $(($sum))





Tips

${} 用于变量替换

$(()) 用于整数运算

Shell 变量都是字符串

数组定义-> array=(…) 元素用空格分割

1. 写一个shell脚本，包含两个数字数组array1和array2，分别初始化为{1，2，3，4，5}和{1，4，9，16，25}。脚本生成并显示一个数组，其中的元素是这两个数组中对应元素的和，数组中第一个元素是1+1=2，第2个元素是2+4=6等。

array1=(1 2 3 4 5)

array2=( 1 4 9 16 25 )

index=0

declare -a new\_array

len=${#array1[@]}

while [ $index -lt $len ]

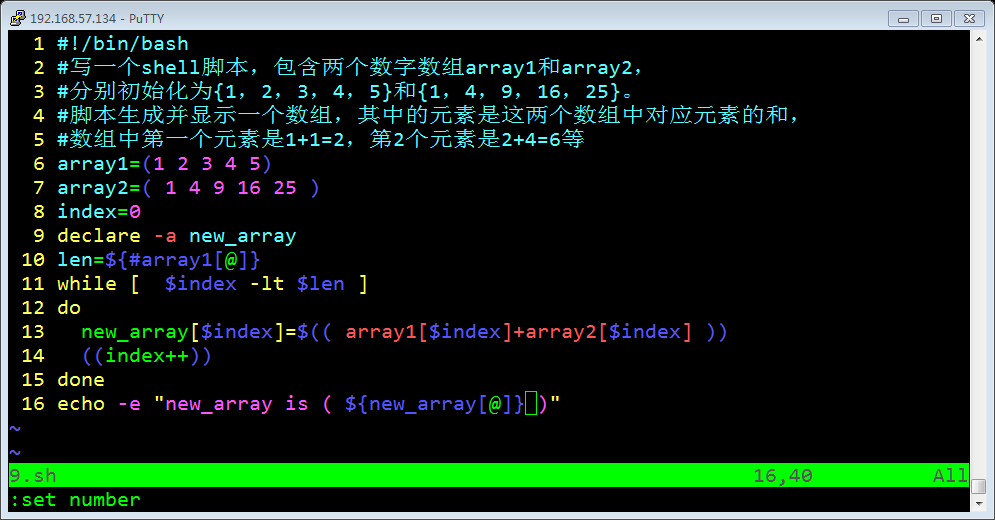
do

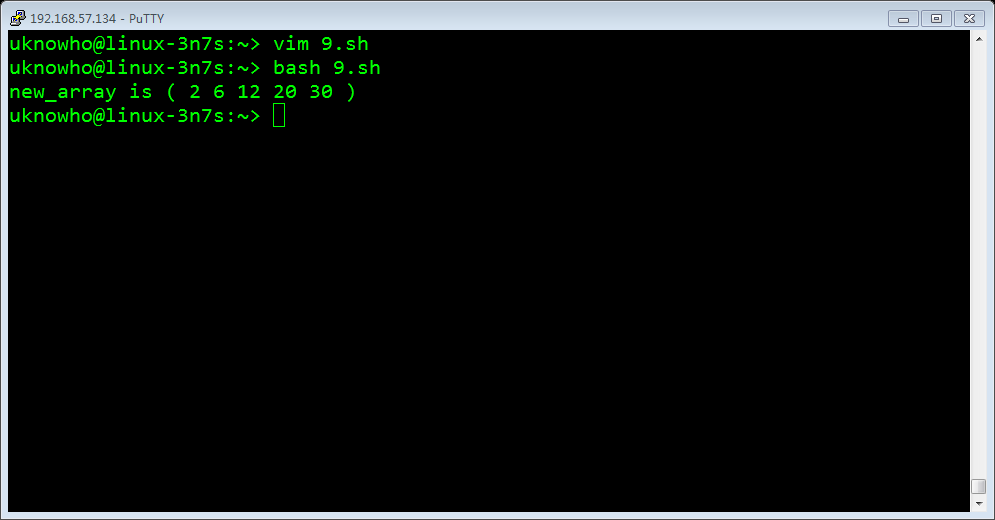
new\_array[$index]=$(( array1[$index]+array2[$index] ))

((index++))

done

echo -e "new\_array is ( ${new\_array[@]} )"





1. 写出一个命令将shell的stdin更改到当前目录下名为data文件，stdout更改到当前目录下名为out的文件。如果data文件包含下面的内容，那么在命令执行后会发生什么？

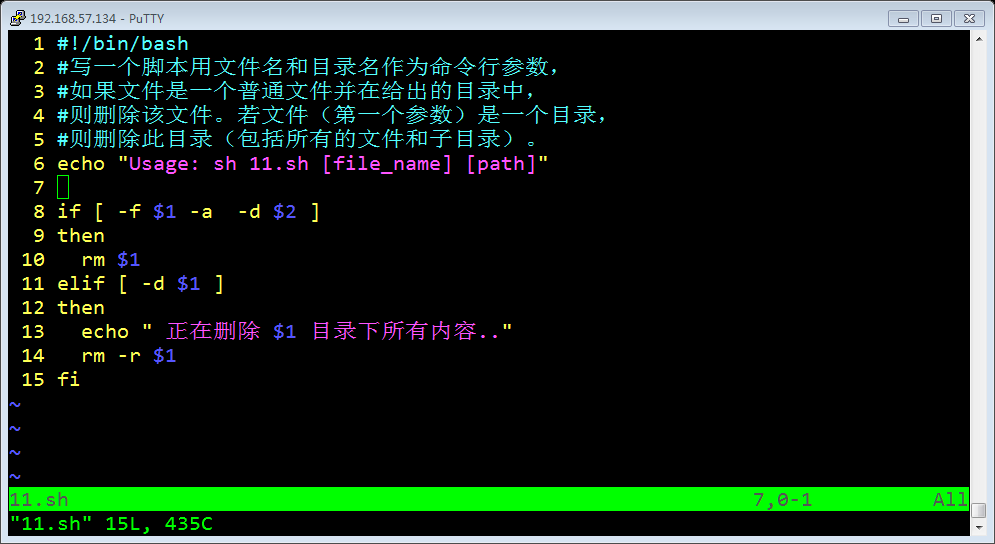
echo –n “The time now is:”

Date

echo –n “The users presently logged on are:”

who

1. 写一个脚本用文件名和目录名作为命令行参数，如果文件是一个普通文件并在给出的目录中，则删除该文件。若文件（第一个参数）是一个目录，则删除此目录（包括所有的文件和子目录）。



#?????????????