**实验四**

1. 实验目的
2. 了解shell的特点和主要种类。
3. 掌握shell脚本的建立何执行方式。
4. 掌握bash的基本语法。
5. 学会编写shell脚本。
6. 实验内容
7. shell脚本的建立和执行。
8. 历史命令和别名定义
9. Shell变量和位置参数、环境变量。
10. Bash的特殊字符
11. 一般控制结构
12. 算数运算及bash函数
13. 主要实验步骤
14. 利用vi建立一个脚本文件。其中包括date、cal、pwd、ls、等常用命令：然后以不同方式执行该脚本
15. 运行history命令、配置历史命令环境
16. 体会bash的命令补齐功能
17. 用alies定义别名，然后执行
18. 对习题4.8中的shell脚本进行编辑，然后执行
19. 对习题4.14要求编写，然后执行
20. 对习题4.18要求编写，然后执行
21. 运行例4.20的程序。若取消其中的“eval”，则会出现什么情况？
22. 实验代码及运行截图

1.利用vi建立一个脚本文件。其中包括date、cal、pwd、ls、等常用命令：然后以不同方式执行该脚本

$ cat first

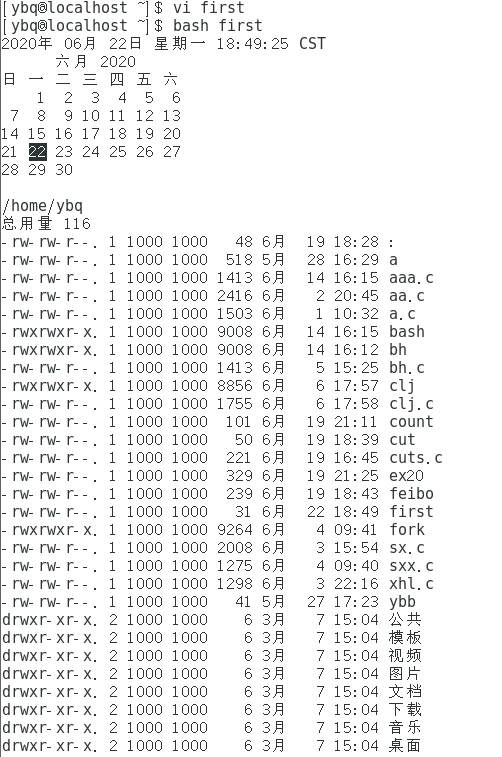
#!/bin/bash

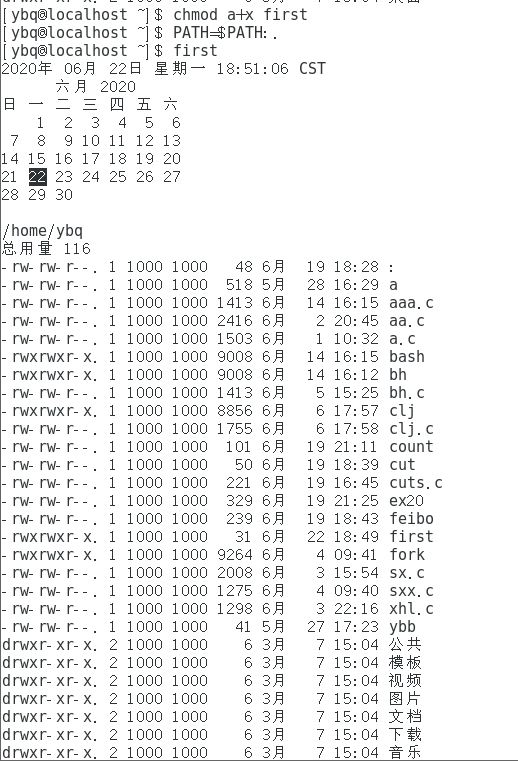
date

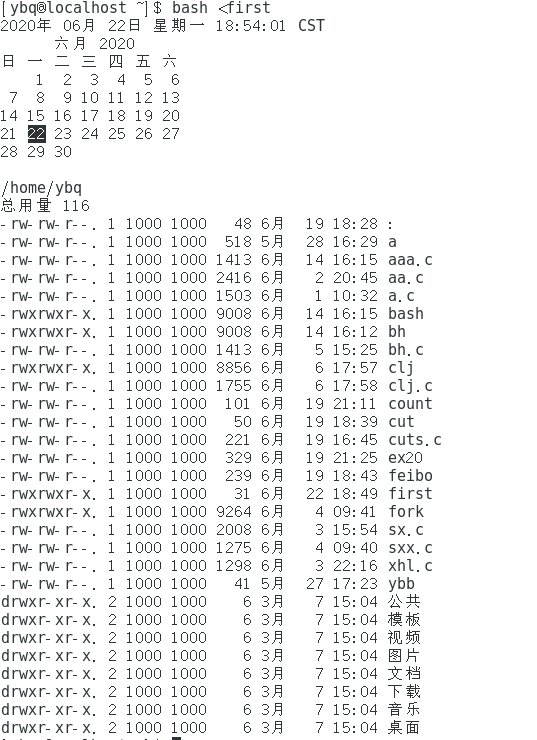
cal

pwd

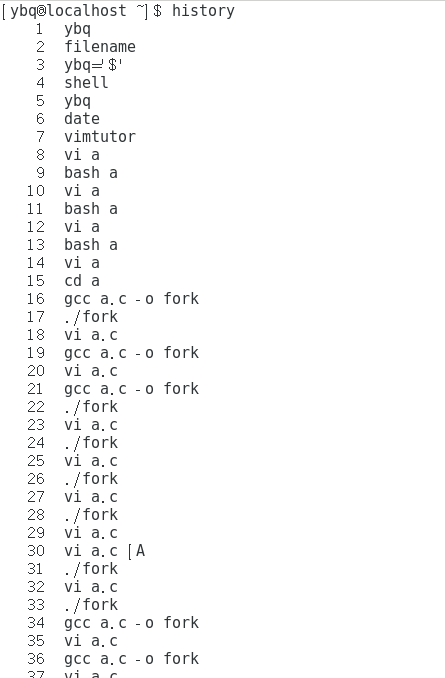
ls -n







1. 运行history命令、配置历史命令环境



在~./bash\_profile文件中新增一行HISTTIMEFORMAT=”%F%T”，显示历史时间

# .bash\_profile

# Get the aliases and functions

if [ -f ~/.bashrc ]; then

. ~/.bashrc

fi

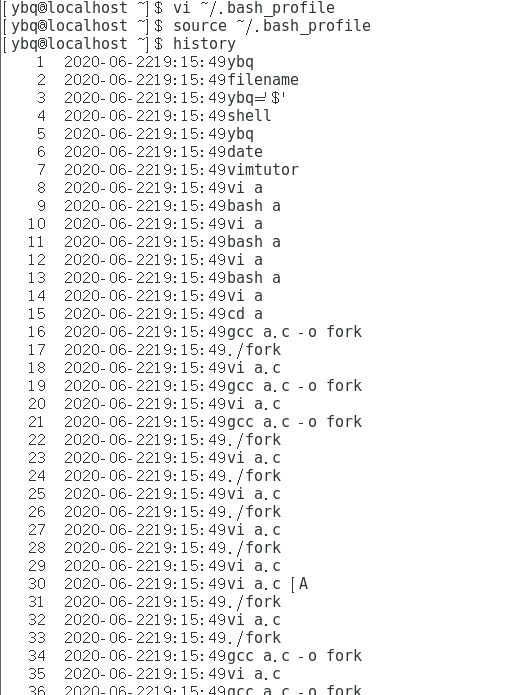
# User specific environment and startup programs

PATH=$PATH:$HOME/.local/bin:$HOME/bin

export PATH

HISTTIMEFORMAT="%F%T"

HISTFILE="~/.second"



3.体会bash的命令补齐功能

4.用alies定义别名，然后执行

# .bashrc

#User specific aliases and functions

alias abc="clear"

# Source global definitions

if [ -f /etc/bashrc ]; then

. /etc/bashrc

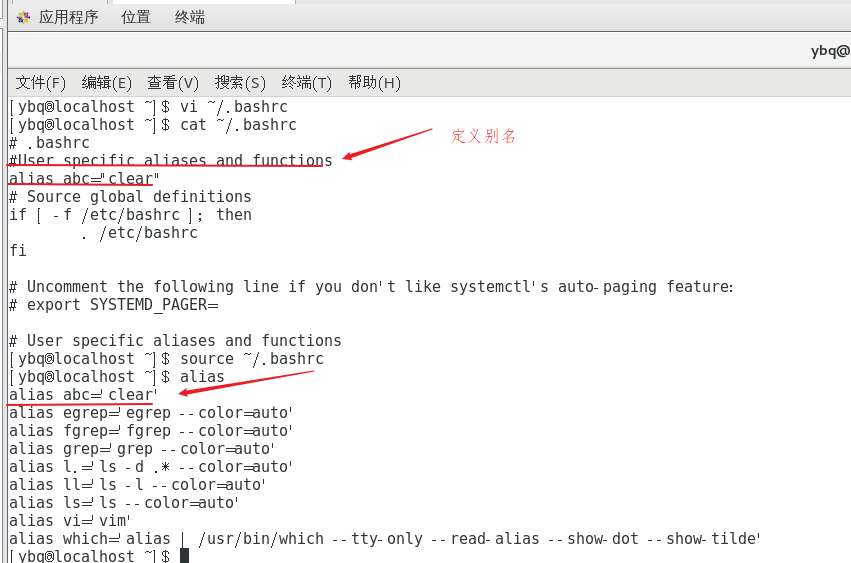
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:

# export SYSTEMD\_PAGER=

# User specific aliases and functions

~



**4.8**

count=$#

cmd=echo

while [ $count -gt 0 ]

do

cmd="$cmd \$$count"

count=`expr $count-1`

done

eval $cmd

count=$#  /\*获取脚本参数个数\*/  
cmd=echo  /\*初始化cmd变量，赋值为"echo"\*/  
while [ Scount -gt 0]  /\*如果count大于0，执行下面的循环\*/  
do  
cmd=' '$cmd \$$count"  /\*给cmd重新赋值，变为"echo ${count的值}”\*/  
count='expr $count-1'  /\*count减1eval $cmd #执行cmd命令。\*/

**4.14**

#!/bin/bash

num[0]=0

num[1]=1

for (( i=2; i<10; i=i+1 ))

do

num1=${num[i-1]}

num2=${num[i-2]}

num[i]=$(($num1+$num2))

done

# 打印所有元素

echo ${num[\*]}

sum=0

for (( i=0; i<10; i=i+1 ))

do

sum=$(($sum+${num[i]}))

done

echo $sum



**4.18**

#!/bin/bash

read line

expr substr "$line" $1 $2;



**4.20**

#!/bin/bash

getc()

{

stty raw

temp=`dd bs=1 count=1 2>/dev/null`

eval $1='$tmp'

stty cooked

}

press\_any\_key()

{

echo -n "Strike any key to continue..."

getc anychar

}

echo -n "Enter a character:"

getc char

echo

echo "You entered $char"

press\_any\_key char

echo

$ ex20

Enter a character:A

You entered A

Strike any key to continue...B

取消其中的“eval”前运行效果



取消其中的“eval”后运行效果

