


第七章

1、查看当前系统下用户Shell定义的环境变量的值。

命令如下，查看当前用户宿主目录完全路径，当前工作目录的绝对路径，平时的命令提示符，第二行待输入命令提示符，默认命令搜索路径，终端类型以及用户识别号

```
echo $HOME
echo $PWD
echo $PS1
echo $PS2
echo $PATH
echo $TERM
echo $UID
```



```
py@CentOS7-64-VM:~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[py@CentOS7-64-VM ~]$ echo $HOME
/home/py
[py@CentOS7-64-VM ~]$ echo $PWD
/home/py
[py@CentOS7-64-VM ~]$ echo $PS1
\u@\h \W \ $
[py@CentOS7-64-VM ~]$ echo $PS2
>
[py@CentOS7-64-VM ~]$ echo $PATH
/usr/local/bin: /usr/local/sbin: /usr/bin: /usr/sbin: /bin: /sbin: /home/py/.local/bin
: /home/py/bin
[py@CentOS7-64-VM ~]$ echo $TERM
xterm-256color
[py@CentOS7-64-VM ~]$ echo $UID
1000
[py@CentOS7-64-VM ~]$
```

2、使用until语句创建一个Shell程序，其功能为计算1~10的平方。

脚本[square.sh](#)

```
#!/bin/bash
var=1
until [ $var -gt 10 ]
do
    sq=`expr $var \* $var`
    echo $sq
    var=$(( $var+1 ))
done
echo "Job completed"
```

出现的问题: []内缺少空格

解答(摘自Stack Overflow)

A bit of history: this is because '[' was historically not a shell-built-in but a separate executable that received the expression as arguments and returned a result. If you didn't surround the '[' with space, the shell would be searching \$PATH for a different filename (and not find it)

第八章

1、使用命令创建用户帐号zhangsan，并设置其口令为111111，设置用户名全称为张三，办公室电话为57111111。

```
sudo useradd zhangsan
sudo passwd zhangsan
usermod -c 张三 zhangsan
chfn zhangsan
Office Phone[]: 57111111
```

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
py@CentOS7-64-VM Linux_Base] $ sudo useradd zhangsan
py@CentOS7-64-VM Linux_Base] $ password zhangsan
ash: password: 未找到命令...
py@CentOS7-64-VM Linux_Base] $ passwd zhangsan
passwd: 只有根用户才能指定用户名。
py@CentOS7-64-VM Linux_Base] $ sudo passwd zhangsan
更改用户 zhangsan 的密码。
新的 密码：
无效的密码： 密码是一个回文
重新输入新的 密码：
passwd: 所有的身份验证令牌已经成功更新。
py@CentOS7-64-VM Linux_Base] $
```

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
[root@CentOS7-64-VM py] # usermod -c 张三 zhangsan
[root@CentOS7-64-VM py] # chfn zhangsan
Changing finger information for zhangsan.
名称 [张三]:
办公 []:
办公电话 []: 57111111
住宅电话 []:

Finger information changed.
[root@CentOS7-64-VM py] #
```

2、使用命令删除用户帐号zhangsan，并且在删除该用户的同时一起删除其主目录。

```
ls /home
```

此时有两个用户，py和zhangsan

```
userdel -r zhangsan
ls /home
```

此时只有用户py

```
py@CentOS7-64-VM:/home/py
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@CentOS7-64-VM py] # ls /home
y zhangsan
root@CentOS7-64-VM py] # userdel -r zhangsan
root@CentOS7-64-VM py] # ls /home
y
root@CentOS7-64-VM py] # █
```

3、使用命令创建组群group1，并且在创建时设置其GID为1800。

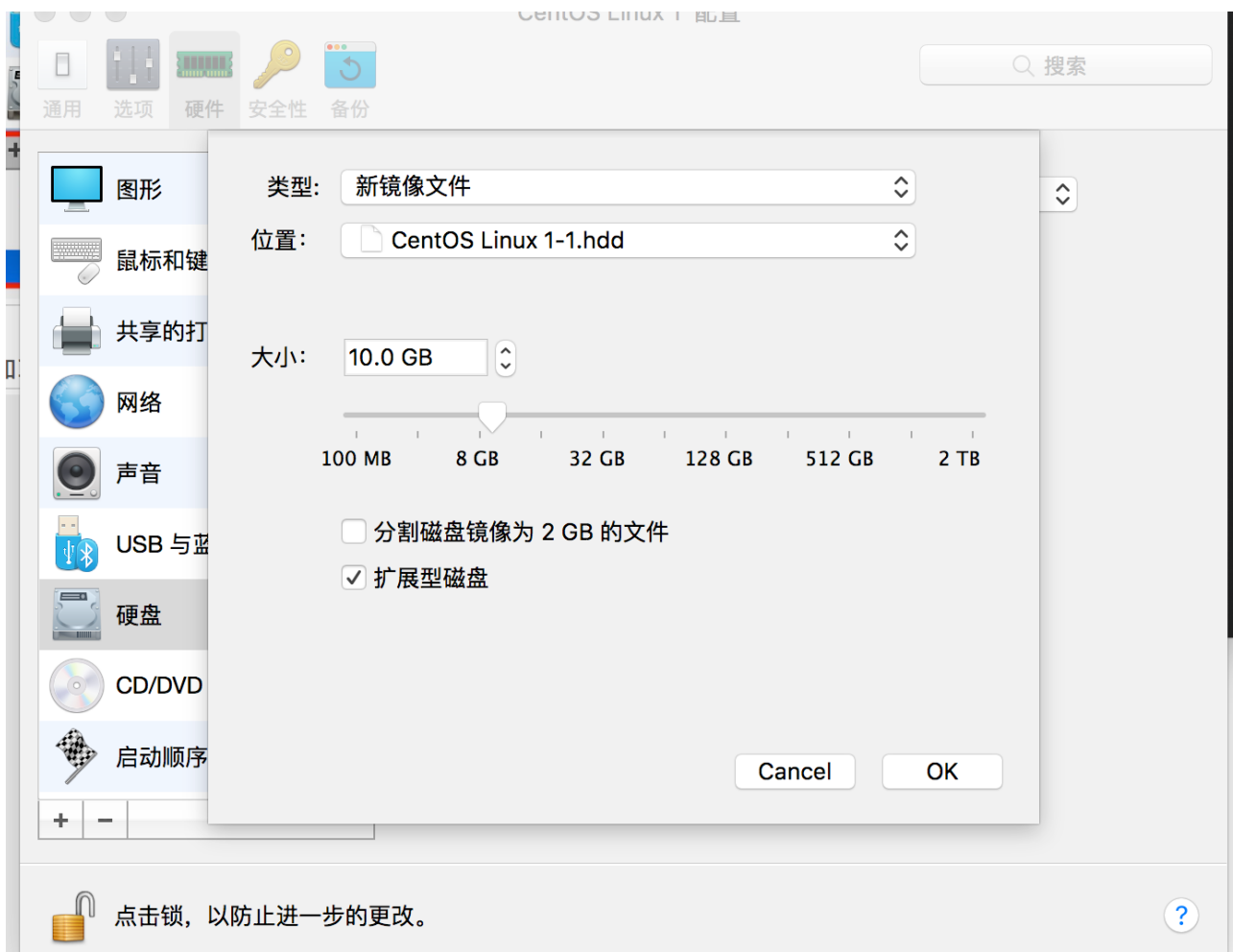
```
groupadd -g 1800 group1
cat /etc/group|grep group1
```

```
py@CentOS7-64-VM:/home/py
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[ root@CentOS7-64-VM py] # groupadd -g 1800 group1
[ root@CentOS7-64-VM py] # cat /etc/group|grep group1
group1:x:1800:
[ root@CentOS7-64-VM py] # █
```

第九章

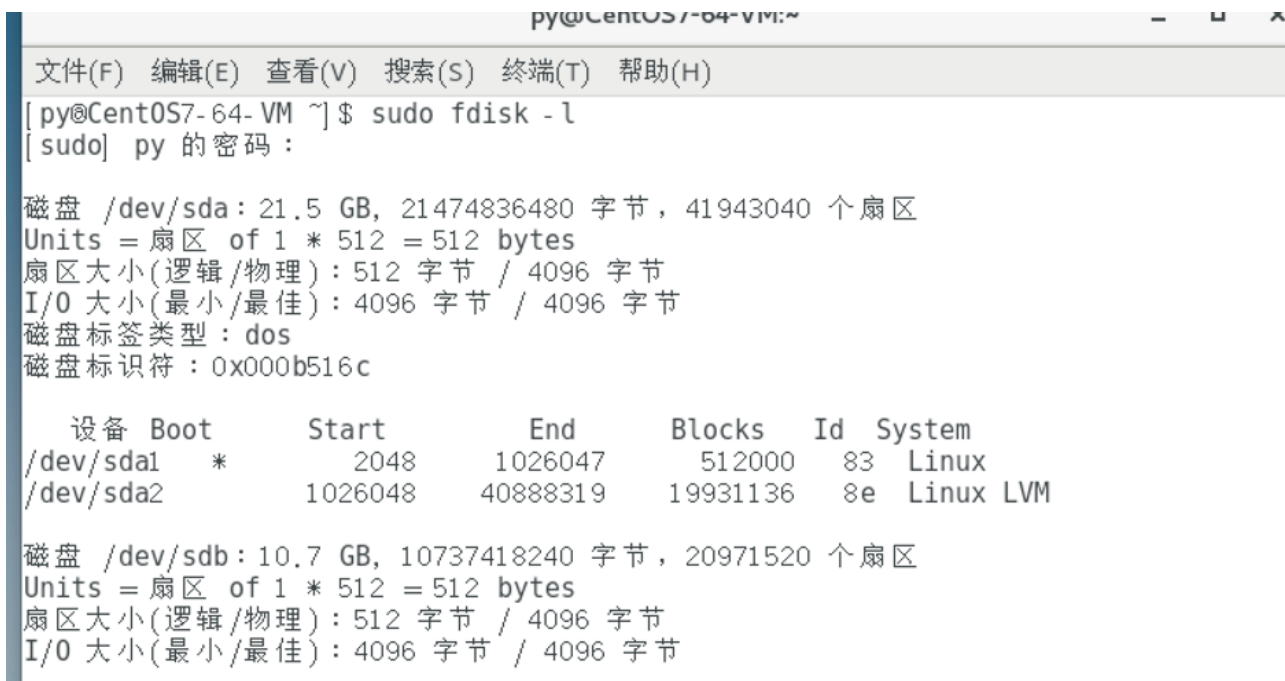
1、在虚拟机上添加一块容量为10GB的新硬盘，对其进行分区，容量分别为1GB和3GB。

1. 先关闭centos7，在Mac parallel界面加入一块新的大小为10G的硬盘



使用命令查看硬盘

```
sudo fdisk -l
```



2. 对默认新磁盘/dev/sdb分区

```
sudo fdisk /dev/sdb
```

一路按提示划分两个区

```

n   add a new partition
o   create a new empty DOS partition table
p   print the partition table
q   quit without saving changes
s   create a new empty Sun disklabel
t   change a partition's system id
u   change display/entry units
v   verify the partition table
w   write table to disk and exit
x   extra functionality (experts only)

```

命令(输入 m 获取帮助): n

Partition type:

```

p   primary (0 primary, 0 extended, 4 free)
e   extended

```

Select (default p): p

分区号 (1-4, 默认 1): 1

起始 扇区 (2048-20971519, 默认为 2048):

将使用默认值 2048

last 扇区, +扇区 or +size{K,M,G} (2048-20971519, 默认为 20971519): +1G

分区 1 已设置为 Linux 类型, 大小设为 1 GiB

命令(输入 m 获取帮助): n

Partition type:

3. 划分结束按p查看分区是两个盘, 1G和3G

命令(输入 m 获取帮助): n

Partition type:

```

p   primary (1 primary, 0 extended, 3 free)
e   extended

```

Select (default p): p

分区号 (2-4, 默认 2):

起始 扇区 (2099200-20971519, 默认为 2099200):

将使用默认值 2099200

last 扇区, +扇区 or +size{K,M,G} (2099200-20971519, 默认为 20971519): +3G

分区 2 已设置为 Linux 类型, 大小设为 3 GiB

4. 按w保存分区设置

设备	Boot	Start	End	Blocks	Id	System
dev/sdb1		2048	2099199	1048576	83	Linux
dev/sdb2		2099200	8390655	3145728	83	Linux

命令(输入 m 获取帮助): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

正在同步磁盘。

lv@CentOS7-64-VM ~1 \$ █

2、对1GB分区创建文件系统为ext4，并将其以只读的方式挂载到/mnt/kk目录中。

先查看一下，今天的分区也是一片祥和呢😄

```
sudo fdisk -l /dev/sdb
```

```
[py@CentOS7-64- VM ~]$ sudo fdisk -l /dev/sdb

磁盘 /dev/sdb: 10.7 GB, 10737418240 字节, 20971520 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理) : 512 字节 / 4096 字节
I/O 大小(最小/最佳) : 4096 字节 / 4096 字节
磁盘标签类型 : dos
磁盘标识符 : 0x7b4ef23d

   设备 Boot      Start         End      Blocks   Id  System
/dev/sdb1             2048        2099199       1048576    83   Linux
/dev/sdb2        2099200        8390655       3145728    83   Linux
[py@CentOS7-64- VM ~]$
```

```
sudo mkfs -t ext4 /dev/sdb1
```

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[py@CentOS7-64- VM ~]$ mkfs -t ext4 /dev/sdb1
mke2fs 1.42.9 (28-Dec-2013)
mkfs.ext4: 权限不够 while trying to determine filesystem size
[py@CentOS7-64- VM ~]$ clear

[py@CentOS7-64- VM ~]$ sudo mkfs -t ext4 /dev/sdb1
[sudo] py 的密码:
mke2fs 1.42.9 (28-Dec-2013)
Discarding device blocks: 完成
文件系统标签=
OS type: Linux
块大小=4096 (log=2)
分块大小=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
65536 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
第一个数据块=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

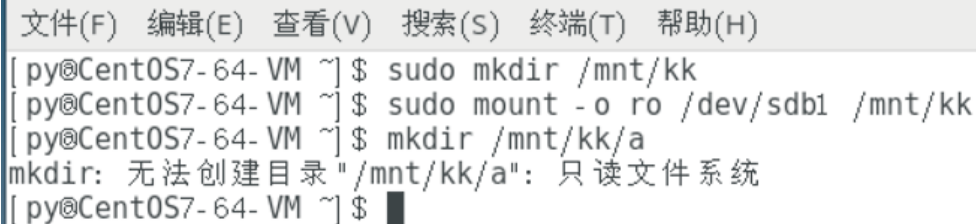
Allocating group tables: 完成
正在写入inode表: 完成
Creating journal (8192 blocks): 完成
Writing superblocks and filesystem accounting information: 完成
```

挂载中

```
mkdir /mnt/kk  
mount -o ro /dev/sdb1 /mnt/kk
```

新建新文件夹测试失败，因为只读

```
mkdir /mnt/kk/a
```



A terminal window with a menu bar (File(F), Edit(E), View(V), Search(S), Terminal(T), Help(H)). The terminal shows the following commands and output:

```
[py@CentOS7-64- VM ~]$ sudo mkdir /mnt/kk  
[py@CentOS7-64- VM ~]$ sudo mount -o ro /dev/sdb1 /mnt/kk  
[py@CentOS7-64- VM ~]$ mkdir /mnt/kk/a  
mkdir: 无法创建目录 "/mnt/kk/a": 只读文件系统  
[py@CentOS7-64- VM ~]$
```

3、修改/etc/fstab文件，使得3GB分区开机时自动挂载到/mnt/jj目录中。

新建文件夹jj

```
sudo mkdir /mnt/jj
```

修改fstab文件

```
sudo vi /etc/fstab
```

在文件末尾加上挂载信息，我这里的3GB分区是/dev/sdb2

```
/dev/sdb2 /mnt/jj ext4 defaults 0 0
```



```
#
# /etc/fstab
# Created by anaconda on Fri Sep 21 22:01:21 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos- root / xfs defaults 0 0
UUID=930ada97-dbc4-490b-b5e0-44210196659c /boot xfs default
ts 0 0
/dev/mapper/centos- home /home xfs defaults 0 0
/dev/mapper/centos- usr /usr xfs defaults 0 0
/dev/mapper/centos- swap swap swap defaults 0 0
/dev/sdb2 /mnt/jj ext4 defaults 0 0
```

4、对/root/ab文件设置权限，所有者为读取、写入和执行权限，同组用户为读取和写入权限，而其他用户没有任何权限。

取消所有用户权限，查看一下，天下大同

```
chmod a- rwx /root/ab
ls -l /root/ab
```

增加拥有者三种权限

```
chmod u+rwx /root/ab
chmod g+rw
ls -l /root/ab
```

```
[py@CentOS7-64- VM ~]$ su root
密码：
su: 鉴定故障
[py@CentOS7-64- VM ~]$ su root
密码：
[root@CentOS7-64- VM py]# chmod a- rwx /root/ab
[root@CentOS7-64- VM py]# ls -l /root/ab
----- . 1 root root 0 8月  8 08:08 /root/ab
[root@CentOS7-64- VM py]# chmod u+rwx /root/ab
[root@CentOS7-64- VM py]# ls -l /root/ab
-rwx----- . 1 root root 0 8月  8 08:08 /root/ab
[root@CentOS7-64- VM py]# chmod g+rw /root/ab
[root@CentOS7-64- VM py]# ls -l /root/ab
-rwxrw--- . 1 root root 0 8月  8 08:08 /root/ab
[root@CentOS7-64- VM py]#
```

5、将文件/root/ab所有者更改为用户lisi。

```
useradd lisi
ls -l /root/ab
chown lisi /root/ab
```

py@CentOS7-64-VM:/home/py

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
[root@CentOS7-64-VM py] # useradd lisi
[root@CentOS7-64-VM py] # ls -l /root/ab
-rwxrw---. 1 root root 0 8月  8 08:08 /root/ab
[root@CentOS7-64-VM py] # chown lisi /root/ab
[root@CentOS7-64-VM py] # ls -l /root/ab
-rwxrw---. 1 lisi root 0 8月  8 08:08 /root/ab
[root@CentOS7-64-VM py] # █
```