用户管理

9:00

就是对passwd、shadow、group三个文件的增删改。

增加一个用户，UID，账户名、家目录、shell程序。

新建的用户，默认的情况下是无法登录的。

创建一个用户的时候，默认情况下，会为其创建一个同名的用户组

创建：

语法：useradd [选项] {UserName}

选项：

-c comment	指定一段注释性描述
-d 目录	指定用户家目录，如果目录不存在，则同时使用-m选项，可以创建家目录。
-g 用户组	指定用户所属的主用户组。
-G 用户附加组	指定用户所属的附加组。
-s Shell程序	指定用户的登录Shell。有一个特殊的shell程序叫nologin，如果某帐号使用此shell，就表示用户不允许登录，该用户通常成为**伪用户**。
-u 用户号	指定用户的用户号

案例：

创建一个普通用户，使用默认值即可。

bash

useradd tom

说明：该账户被创建完之后，会有哪些信息

UID，由于是当前系统的第一个账户，那么它的UID是500。

家目录(主目录)，因为是默认创建，所以它的家目录所在位置是/home/tom

用户组，创建用户的同时，会为其创建一个同名的用户组，tom组

shell程序，默认会是bash

创建一个普通用户，其家目录是在/tedu/jerry，主组root,附加组tom,adm。

bash

useradd -d /tedu/jerry -m -g root -G tom,adm jerry

指定用户的shell程序为/bin/sh，UID为80000

bash

useradd -s /bin/sh -u 80000 test1

说明：虽然CentOS系统中默认的UID是65536个(0-65535)，但是可以手动指定它的UID超出此范围。

测试再创建账户是UID会不会延续上案例的UID。

bash

useradd test2

正常来应该按照上次创建用户的UID往后+1。但是80000已超过系统默认的范围，所以不会根据超出范围后UID进行+1延续。

创建伪用户：

bash

useradd -s /sbin/nologin test3

修改：

语法：usermod [选项] {UserName}

选项：

-c comment	指定一段注释性描述
-d	目录 指定用户主目录，如果目录不存在，则同时使用-m选项，可以创建主目录。
-g 用户组	指定用户所属的用户组。
-G 用户组,用户组	指定用户所属的附加组。如果想要在原有的基础上追加附加组，使用-aG的方式。
-s Shell文件	指定用户的登录Shell。有一个特殊的shell程序叫nologin，如果某帐号使用此shell，就表示用户不允许登录，该用户通常成为伪用户。
-u 用户号	指定用户的用户号

案例：

修改test账户的UID。

bash

usermod -u 9000 test1

修改账户的附加组为test、test1：

bash

usermod -G test,test1 jerry

为账户追加附加组件root、adm

bash

usermod -aG root,adm jerry

将test5的伪用户状态修改为正常普通用户

bash

usermod -s /bin/bash test5

说明：test5之前的shell程序是/sbin/nologin

删除：

语法：userdel [选项] {UserName}

选项：

-r	删除账户的同时删除该账户的家目录

案例：

bash

userdel zhangsan

说明：该命令执行完成之后，在home目录下依然保留了zhangsan的家目录。

userdel -r lisi

说明：该命令之后之后，home目录中的lisi目录会被一并删除，且不保留它的工作文件。

一般来说，公司中删除账户的操作比较少，就算删除账户时，一般也不用-r的选项，因为人虽然离职，但是有可能此人还会回来继续任职~

如果员工离职，完全没有必要删除账户，可以使账户无法登录即可，比如nologin，锁定账户，删除密码等手段都是可以防止资料泄漏。

注：

Windows和Linux系统(其他系统没测试)，没有密码的账户都是不允许通过远程的方式进行访问的。

密码管理：

语法：passwd [选项] {UserName}

选项：

-l	锁定账户
-u	解锁账户
-d	删除密码

passwd 可以不跟选项、用户名，默认是修改自己的帐号密码。

修改他人密码，必须具备管理员权限(并不一定非要是root账户)

普通账户，只能修改自己的密码。

管理员修改他人的密码，不需要满足密码策略（as12AS！@）。

普通账户修改密码时，必须满足密码安全策略。

管理员修改他人密码：

passwd {UserName}

修改自己的密码

passwd

案例：

锁定zhangsan账户

bash

passwd -l zhangsan

解锁zhangsan账户

bash

passwd -u zhangsan

删除zhangsan账户密码

bash

passwd -d zhangsan

案例中，锁定账户和删除账户密码都是可以达到不允许账户远程登录的效果。

用户身份切换：

su {UserName}	表示切换用户之后，依然停留在当前目录
su - {UserName}	表示切换用户之后，去到该用户的家目录

普通用户切换root时，书写格式不需要写成su - root，直接su即可。

案例：

当前位置，在/root下

bash

su zhangsan

该命令执行过后，切换到zhangsan账户，但是路径依然停留在/root目录下

```
[root@localhost ~]# pwd
/root
[root@localhost ~]# su zhangsan
[zhangsan@localhost root]$ pwd
/root
[zhangsan@localhost root]$
```

bash

su - zhangsan

该命令执行过后，切换到zhangsan账户，同时会去到zhangsan账户的家目录下。

```
[root@localhost ~]# su - zhangsan
[zhangsan@localhost ~]$ pwd
/home/zhangsan
[zhangsan@localhost ~]$
```

小总结，su与账户名之间加不加"-"区别在于路径。

shell程序的简单介绍：

10:52

/bin/sh和/bin/bash都是Linux的shell程序，但是表现的形式不同，具体执行功能是一样的，可以理解为bash是sh的升级版，兼容版本

/bin/sh	-sh-4.1$
/bin/bash	[root@localhost tedu]#

从上面的内容可以看得出来，bash确实是比sh提供的内容更多，更为详细。

用户组的增删改

11:41

创建：

语法：groupadd [选项] {groupName}

选项：

-g GID	指定新用户组的GID
-o	通常与-g同时使用，使新用户组可以与系统已有的组ID相同。

系统底层会将两个GID相同用户组识别为同一个用户组，这样做的目的是让两个用户组的权限相同。识别规则，后者遵循前者。

案例：

新建一个用户组

#groupadd group1

此命令向系统中增加了一个新组group1，新组的组标识号，在当前已有的最大组标识号的基础上加1

groupadd -g 101 group2

此命令向系统中增加了一个新组group2，同时指定新组的组标识号是101。

创建一个普通的用户组，并将该组添加为tom账户的附加组

bash

usermod -aG group1 tom

创建一个与上面案例GID相同的用户组

bash

groupadd -g 1000 -o group2

修改：

语法：groupmod [选项] {groupName}

选项：

-g GID	指定新用户组的GID
-o	通常与-g同时使用，使新用户组可以与系统已有的组ID系统。
-n	用来修改组ID　group -n newGroupName oldGroupName

案例：

将组group2的组标识号修改为102。

groupmod -g 102 group2

新建一个group3，并将其组名修改成big1902

bash

groupadd group3

groupmod -n big1902 group3

删除：

语法：groupdel {groupName}

案例：

bash

groupdel big1902

注：

如果删除的用户组，已经被用户追加为附加组，对应的所有用户的该附件组会被撤销掉。

如果被删除的用户组，已经被用户指定为主组，则该用户组无法被删除。(可以理解为像Windows中文件被占用时不能被删除。)

用户组被设置为主组不能删除，普通组以及附加组都可以被删除。

用户组的切换：

某用户属于多个用户组时，想要访问其他用户组中的内容时，必须切换用户组才行。

newgrp {GroupName}

前提是该账户确实是拥有多个用户组。

passwd、shadow、group文件详解

11:57

passwd:

root:x:0:0:root:/root:/bin/bash

passwd这个文件的每行内容由冒号隔开，分为7段

第一段	账户名	不要使用：. - + /
第二段	密码	passwd这个文件是所有人都可查看，所以密码虽然是加密的，但是依然不安全。
第三段	UID	系统用来标识内部的账户，通常UID和账户是对应的关系。如果出现了两个不同账户名但是却使用的相同的UID，那么系统就将它们两个识别为同一个账户，只不过它们拥有不同的账户名、shell程序、家目录。后者遵循前者。
第四段	GID	此处记录的是该账户的主组信息。
第五段	注释描述	对账户的描述信息，通常自定义账户都不写这个。
第六段	家目录	用户登录系统之后的默认工作空间，该空间除root和本人意外，默认是不允许其他人访问。
第七段	Shell程序	Linux、类Unix系统中的特有程序。

shadow：

mengxb:1P3i2zUCl$QxgXUAArJQhNWFxl0EFaD1:17365:0:99999:7:::

第一段	用户名
第二段	加密后的口令(注：不允许手动修改密码内容,如果含有不属于集合 { $./0-9A-Za-z }中的字符，则对应的用户不能登录)
第三段	从密码创建至今的天数。从1970年1月1日开始计算值。
第四段	表示上次和下次修改密码之间的间隔，如果是0表示无间隔限制
第五段	自密码创建时刻起，最大的有效期（天）
第六段	密码到期前的N天，提醒用户修改密码。
第七段	缺省值为空，该字段允许密码到期之后N天之内还依然可以登录。
第八段	缺省值为空，该字段表示一个绝对的天数，意为到期之后不允许登录，也可以理解为密码的有效存活期。

group:

root:x:0:jerry

第一段	用户组名
第二段	组的密码，通常用x或者*来表示。部分系统中没有组密码的设定
第三段	GID
第四段	组内的成员列表。文件中显示的账户名都是将该组作为附加组。如果是主组，在此不给与显示。

说明：

root账户和root用户组是两码事，某个账户就算是加入了root组，那么也不代表它具备root权限。

因为在Linux系统，默认情况下，用户的权限高于用户组。

权限

14:42

权限

在Linux系统对于权限的设定非常的敏感，如果某个用户执行一个操作时，提示权限不足，那么根据Linux系统的权限设定的思想(没有权限绝对不会睁一只眼闭一只眼)，就能够判断出该用户不具备此文件的执行权限。

在Linux系统中，有以下的权限表示。业内人士称之为：

逻辑权限

物理权限

普通用户的root的权限。

逻辑权限：

在Linux系统中不管是文件还是目录。(在Linux系统中，将所有的东西都视为文件。)都有固定权限表示。

例：

```
drwxr-xr-x. 2 root root  4096 5月  13 15:27 home
-rw-r--r--. 1 root root 45537 5月  13 11:15 install.log
```

两个文件分别是：第一个是目录，第二个是普通文件

根据信息的**第一个字母(文件类型)**来查看，d表示该文件是一个目录文件，-表示该文件是一个普通文件。

后面每三个权限成为一组，每组中分别有三个权限：

字符	权限	数字
r	读	4
w	写	2
x	执行	1

除了第一个字母不参与权限的表示，其他的都为权限标识符。

每三个为一组，共有三组：

第一组	用户	user
第二组	用户组	group
第三组	其他人	other

说明：Linux系统中，不管是什么系统，权限的标识符号的位置是不会发生任何的改变，也就是说，
第1个永远是文件类型，
第2-4(第一组)永远是读、写、执行，用户的权限，
第5-7(第二组)永远是读、写、执行，用户组的权限，
第8-10(第三组)永远是读、写、执行，其他人的权限。

如果某个文件权限标识为----------，那么则说明此文件不允许任何的读取、写入、执行

修改文件/夹的权限：

chmod命令可以用来修改某个文件或文件夹的权限。

选项

-R	递归处理

修改文件/夹的权限时，可以使用字符权限，也可以使用数字权限。

案例：

bash

touch test_1 # 当前文件的权限是-rw-r--r--

将此文件的权限修改为----------

bash

chmod 000 test_1

为此文件，每组都增加一个读的权限

bash

chmod 444 test_1

or

chmod +r test_1

a=all,u=user,g=group,o=other

为此文件的用户增加一个rw-，组增加一个r-x，其他人---。

chmod u+rw,g+rx test_1

or

chmod 650 test_1

0表示没有权限

chmod在修改文件权限的时候，哪个便捷用哪个方法。

比如：

如果要是给三组增加执行权限的时候，+x就数字计算要快。字符权限就比数字要便捷(不需要计算)

如果是为每组增加不同权限的时候，用数字比较便捷(书写便捷)。

物理权限：

修饰某个文件/夹不允许被修改。注意：不能给/ p/tm /dev /var 加保护

即便是root权限也不一定所有的文件都可以删

chattr [选项] file/dir

选项：

i	表示不能以任何方式进行文件/夹的修改，增加，删除
a	表示文件/夹只能追加，不能修改，删除 >>(追加) >(覆盖)
+ <属性>	表示开启某文件/夹的权限
- <属性>	表示关闭某文件/夹的权限

R	表示递归处理。

案例：

```
# bash
    touch big1902
    chattr +i big1902    # 表示该文件不允许修改，删除，增加。

    touch big 1902_1
    chattr +a big1902_1 # 表示该文件只允许追加内容，不允许删除和修改。
```

a、i的使用场景：

通常情况，log文件用a的属性。如果是cfg（配置文件）文件用i的属性。

lsattr 查看文件的物理权限(属性)

lsattr [选项] 文件/夹

选项：

R	表示递归处理
a	表示查看所有文件的属性，包括隐藏
d	显示目录的属性，而不是目录下的文件的属性

修改用户的所有者（属主）和属组

chown root /u　　将 /u 的属主更改为"root"。
chown :staff /u　　将/u的属组更改为staff
chown root:staff /u和上面类似，但同时也将其属组更改为"staff"。

普通用户的超级权限：

sudo(SuperUser Do)，它可以让普通用户执行root的权限。sudo可以限制用户执行部分root的权限。

sudo会记录用户执行过的每一条命令，便于查阅服务起出事之前的状态。

好处：

使用自己配置好的用户环境
不需要知道root密码，保证root的密码安全
可以限制用户执行有限的root权限
sudo执行的每条命令都会被记录，便于日后的日志审计，例如用户执行过高危操作命令。
在/etc/sudoers文件中配置 Defaults　logfile =/home/log.txt 即可

sudoers文件解释

```
## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##该文件允许特定用户像root用户一样使用各种各样的命令，而不需要root用户的密码
##
## Examples are provided at the bottom of the file for collections
```

of related commands, which can then be delegated out to particular
users or groups.
在文件的底部提供了很多相关命令的示例以供选择，这些示例都可以被特定用户或
用户组所使用
This file must be edited with the 'visudo' command.
该文件必须使用"visudo"命令编辑
Host Aliases
#主机别名
Groups of machines. You may prefer to use hostnames (perhap using
wildcards for entire domains) or IP addresses instead.
对于一组服务器，你可能会更喜欢使用主机名（可能是全域名的通配符）
或IP地址代替，这时可以配置主机别名

Host_Alias   FILESERVERS = fs1, fs2
Host_Alias   MAILSERVERS = smtp, smtp2
User Aliases
#用户别名
These aren't often necessary, as you can use regular groups
(ie, from files, LDAP, NIS, etc) in this file - just use %groupname
rather than USERALIAS
这并不很常用，因为你可以通过使用组来代替一组用户的别名
User_Alias ADMINS = jsmith, mikem

Command Aliases
These are groups of related commands...
指定一系列相互关联的命令（当然可以是一个）的别名，通过赋予该别名sudo权限，
可以通过sudo调用所有别名包含的命令，下面是一些示例

Networking
#网络操作相关命令别名
Cmnd_Alias NETWORKING = /sbin/route, /sbin/ifconfig, /bin/ping, /sbin/dhclient,
 /usr/bin/net, /sbin/iptables, /usr/bin/rfcomm, /usr/bin/wvdial, /sbin/iwconfig,
 /sbin/mii-tool
Installation and management of software
#软件安装管理相关命令别名
Cmnd_Alias SOFTWARE = /bin/rpm, /usr/bin/up2date, /usr/bin/yum
Services
#服务相关命令别名
Cmnd_Alias SERVICES = /sbin/service, /sbin/chkconfig
Updating the locate database

```
#本地数据库升级命令别名
Cmnd_Alias LOCATE = /usr/sbin/updatedb
## Storage
#磁盘操作相关命令别名
Cmnd_Alias STORAGE = /sbin/fdisk, /sbin/sfdisk, /sbin/parted, /sbin/partprobe,
/bin/mount, /bin/umount
## Delegating permissions
#代理权限相关命令别名
Cmnd_Alias DELEGATING = /usr/sbin/visudo, /bin/chown, /bin/chmod, /bin/chgrp
## Processes
#进程相关命令别名
Cmnd_Alias PROCESSES = /bin/nice, /bin/kill, /usr/bin/kill, /usr/bin/killall
## Drivers
#驱动命令别名
Cmnd_Alias DRIVERS = /sbin/modprobe
#环境变量的相关配置
# Defaults specification
#
# Disable "ssh hostname sudo <cmd>", because it will show the password in clear.
#         You have to run "ssh -t hostname sudo <cmd>".
#
Defaults    requiretty
Defaults    env_reset
Defaults    env_keep = "COLORS DISPLAY HOSTNAME HISTSIZE INPUTRC KDEDIR \
                    LS_COLORS MAIL PS1 PS2 QTDIR USERNAME \
                    LANG LC_ADDRESS LC_CTYPE LC_COLLATE LC_IDENTIFICATION \
                    LC_MEASUREMENT LC_MESSAGES LC_MONETARY LC_NAME LC_NUMERIC \
                    LC_PAPER LC_TELEPHONE LC_TIME LC_ALL LANGUAGE LINGUAS \
                    _XKB_CHARSET XAUTHORITY"
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## 下面是规则配置：什么用户在哪台服务器上可以执行哪些命令（sudoers文件可以在多个系统上
共享）
## Syntax:
##语法
##      user    MACHINE=COMMANDS
##  用户 登录的主机=（可以变换的身份） 可以执行的命令
##
## The COMMANDS section may have other options added to it.
```

命令部分可以附带一些其它的选项
##
Allow root to run any commands anywhere
允许root用户执行任意路径下的任意命令
root   ALL=(ALL)   ALL
Allows members of the 'sys' group to run networking, software,
service management apps and more.
%sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS
允许sys中户组中的用户使用NETWORKING等所有别名中配置的命令

Allows people in group wheel to run all commands
%wheel   ALL=(ALL)   ALL
允许wheel用户组中的用户执行所有命令
Same thing without a password
允许wheel用户组中的用户在不输入该用户的密码的情况下使用所有命令
%wheel   ALL=(ALL)   NOPASSWD: ALL
Allows members of the users group to mount and unmount the
cdrom as root
允许users用户组中的用户像root用户一样使用mount、unmount、chrom命令
%users ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
Allows members of the users group to shutdown this system
%users localhost=/sbin/shutdown -h now
允许users用户组中的用户像root用户一样使用shutdown命令

实际案例演示
实例1：让普通用户fieldyang具有/etc/init.d/nagios脚本重启的权限，可以在/etc/sudoers添加如下设置：

[root@test ~]# visudo
fieldyang ALL=NOPASSWD:/etc/init.d/nagios restart

实例2：让普通用户fieldyang具有所有超级用户的权限而又不用输入密码
[root@test ~]# visudo
fieldyang ALL=（ALL)NOPASSWD:ALL
[fieldyang@test ~]#sudo su -
[fieldyang@test ~]#pwd
/root

实例3：针对MySQL数据库的设置，让test组中的test用户具备/etc/init.d/mysqld的权限

######################### mysql ################

1.

[root@test ~]# groupadd test

[root@test ~]# useradd -g test -m -d /home/test -s /bin/bash test

[root@test ~]# passwd test

2.

[root@test ~]# visudo

test ALL=(ALL) NOPASSWD: /etc/init.d/mysqld

test ALL=(ALL) /etc/init.d/mysqld

3. start/stop mysql

3.1) start mysql

login test

[root@test ~]# su test

[test@test ~]$ sudo /etc/init.d/mysqld start

3.2) stop mysql

login test

[root@test ~]# su test

[test@test ~]$ sudo /etc/init.d/mysqld stop

实例4：针对tomcat的设置，让test组中的test用户具备tomcat操作的权限

######################### tomcat ################

1.

[root@test ~]# groupadd test

[root@test ~]# useradd -g test -m -d /home/test -s /bin/bash test

[root@test ~]# passwd test

2.

[root@test ~]# visudo

test ALL=(ALL) /usr/local/tomcat/bin/shutdown.sh,/usr/local/tomcat/bin/startup.sh

test ALL=(ALL) NOPASSWD:
/usr/local/tomcat/bin/shutdown.sh,/usr/local/tomcat/bin/startup.sh

3.

[root@test ~]# vim /usr/local/tomcat/bin/catalina.sh

JDK

export JAVA_HOME=/usr/local/jdk

export JRE_HOME=$JAVA_HOME/jre

4. start/stop tomcat

4.1) start tomcat

login test

[root@test ~]# su test

[test@test ~]$ sudo /usr/local/tomcat/bin/startup.sh

[test@test ~]$ ss -ntlup | grep java

[test@test ~]$ curl -I http://localhost:8080

4.2) stop tomcat

login test

[root@test ~]# su test

[test@test ~]$ sudo /usr/local/tomcat/bin/shutdown.sh

练习

15:00

练习1

需求：现在大数据部门有一个加密狗文件。大数据部门的人都可以去连接它给大家上课，但是别的部门不行

1，在/home目录下touch 加密狗文件（Usbkey）；

touch UsbKey

2，创建大数据组（Bigdata）；

groupadd bigdata

3，修改加密狗文件的所有者为root和大数据组；

chown root：bigdata UsbKey

4，修改加密狗文件的权限为：root用户可读可写可执行，大数据组的成员可读可写可执行，其他人没有权限；

chmod 770 UsbKey

4，编辑加密狗文件实现其功能（写一句话就行）；

5，创建大数据组的成员，并将其加入到大数据组；

useradd -g bigdata caolaoshi

useradd -g bigdata piaolaoshi

useradd xiaolaoban

通过不同权限的人，来观察加密狗的使用情况

练习2

需求：现在1910班有两个组，一个是A组，另外一个是B组。

其中A组成员有zhangsan，lisi。B组成员有wangwu，zhaoliu。

A组成员的家目录在/big1910/A/目录下，且只有A组的人可以进

B组成员的家目录在/big1910/B/目录下，且只有B组的人可以进

现在要实现，组内之间资料共享，其他人无权查看。

资料如下：A组有两本书：

book1：所有者和属组为root用户和A组。权限为root用户可读可写可执行，A组的人可读可写可执行，其他人没有权限。

book2：所有者和属组为root用户和A组，权限为root用户可读可写可执行，A组的可读可执行不可写，其他人没有权限。

#创建用户家目录

mkdir -p /big1910/A

mkdir /big1910/B
#创建用户并指定家目录
useradd -d /big1910/A/zhangsan -m zhangsan
useradd -d /big1910/A/lisi -m lisi
useradd -d /big1910/B/wangwu -m wangwu
useradd -d /big1910/B/zhaoliu-m zhaoliu

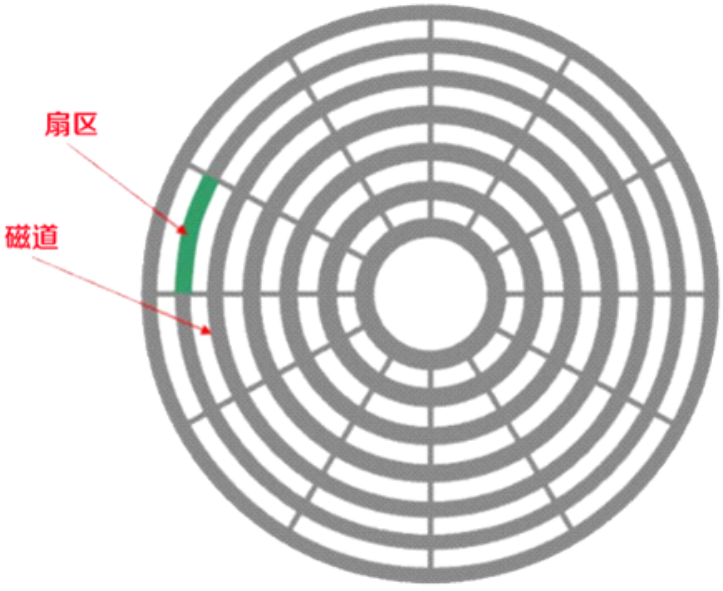
#创建用户组：
groupadd -g 1000 A
groupadd -g 1001 B
#将用户分别加入对应的组内
usermod -g A zhangsan
usermod -g A lisi
usermod -g B wangwu
usermod -g B zhaoliu

既然是组内的成员进行共享，那么就应该可以使用权限的方式来实现，先修改组的目录权限为其他人无权限
#cd /big1910
chmod 750 A
chmod 750 B
通过ll命令我们能看到A组和C组所有者为root用户和root组。
所以修改A目录的所有者为A，B组的所有者为B
chown root:A A
chown root:B B

在A组目录中touch book1，并修改权限为770，所有者为root:A，同时创建book2，修改权限为750，所有者为root:A

磁盘、文件系统

15:58

- 磁盘，它是由一个个盘片组成的，从盘片的结构上来看 图中的一圈圈灰色同心圆为一条条磁道，从圆心向外画直线 ，可以将磁道划分为若干个弧段，每个磁道上一个弧段被称为一个扇区（图中绿色部分）。扇区是磁盘的最小组成单 元，常是512字节。
- 

- 磁盘分区：指定分割区域起始与结束磁柱
- 磁盘存取的区域 例如A磁柱到B磁柱之间的区块，磁盘在此分割区域内操作系统能够知道它可以在指定区块进行文件读，写，查询等操作
- 但是需要注意使用硬盘之前需要格式化！
- 因为每种操作系统所设定的文件属性/权限，以及存放数据的格式 有所不同

- **Ext2/Ext3/Ext4区别**
- Ext*、NTFS和FAT32这三个都是文件系统格式
- Linux kernel自2.6.28开始正式支持新的文件系统Ext4
- Ext4是Ext3的改进版，修改了Ext3中部分重要的数据结构
- Ext3对Ext2，只是增加了一个日志功能
- Ext4可以提供更佳的性能和可靠性，还有更为丰富的功能，更大的文件系统和更大的文件。
  - 较之Ext3所支持的最大16TB文件系统和最大2TB文件，Ext4分别 支持1EB（1,048,576TB，1EB=1024PB，1PB=1024TB）的文件系统，以及16TB的文件。
- **无限数量的子目录**
  - Ext3只支持32,000个子目录，而Ext4支持理论值的无限数量的子目录

  延迟分配
  - Ext3的数据块分配策略是尽快分配，而Ext4是尽可能地延迟分配，直 到文件在cache中写完才开始分配数据块并写入磁盘。
  - 如此能优化整个文件的数据块分配，显著提升性能。

  快速fsck（文件系统检查）
  - 老的fsck会很慢，因为它要检查所有的索引节点(inode)
  - Ext4给每个组的索引节点表中添加了一份未使用inode的列表，执行fsck就可以跳过它们而只去检查那些在用的索引
  -

  持久预分配（Persistentpreallocation）
- 常常会预先创建 一个与所下载文件大小相同的空文件，以免未来的数小时或数天 之内磁盘空间不足导致下载失败。Ext4在文件系统层面实现了持久预分配并提供相应的API，比应用软件自己实现更有效率。
-

eclipse-jee-oxygen-3a-win32-x86_64.zip.xltd	2018/5/16 1:25	迅雷临时数据文件	341,896 KB
eclipse-jee-oxygen-3a-win32-x86_64.zip.xltd.cfg	2018/5/16 1:25	CFG 文件	4 KB

SWAP（交换分区）概述

- 使用磁盘来存储内存不够而“溢出来”的内容(拿硬盘空间来存储内存“溢出”的数据）。
- 当系统的物理内存不够用的时候，就需要将物理内存中的一部分 空间释放出来，以供当前运行的程序使用。
- 最容易成为被释放的对象：一些很长时间没有什么操作的程序。 –被 保存到Swap空间中。等到那些被换出的程序要继续运行时，再从Swap中恢复保存的数据到内存中。
- 一般来说可以按照如下规则设置swap大小：

  •8G以内的物理内存，SWAP 设置为内存的2倍。

•8G-16G以内的的物理内存，SWAP 等于内存大小或者设置为8G。

•16G-256G 的物理内存，SWAP 设置为实际内存的1/2即可。

- **系统什么时候会使用swap？**
- 实际上，并不是等所有的物理内存都消耗完毕之后，才去使用 swap的空间，什么时候使用是由 swappiness 参数值控制。
- [root@localhost ~]# cat /proc/sys/vm/swappiness
- 60
- [root@localhost ~]#
- 默认值是60。swappiness=0的时候表示最大限度使用物理内存 ，然后才是 swap空间，swappiness = 100的时候表示积极的使 用swap分区，并且把内存上的数据及时的搬运到swap空间里面。
- **如何修改swap参数**
- 临时性修改：
  - [root@localhost ~]# sysctl vm.swappiness=10
  - [root@localhost ~]# cat /proc/sys/vm/swappiness
- 10
- 这里我们的修改已经生效，但是如果我们重启了系统，又会变成60.
- 永久修改：
- 在/etc/sysctl.conf 文件里添加如下参数： vm.swappiness=10

挂载

16:37

为什么要挂载，因为文件系统并不能够直接使用。

Windows的文件系统需要盘符来表示

Linux的文件系统需要目录作为入口。

分区的格式就是文件系统。

挂载：

mount 文件系统 目录(挂载点)

案例：

挂载光盘镜像文件

bash

mkdir /home/cdrom # 此处创建目录cdrom并不是非要这个名称，是因为想做到见名知意。

mount /dev/cdrom /home/cdrom

挂载U盘：

需要注意：U盘的格式如果为NTFS，那么需要安装一个插件之后才能够进行挂载，否则无法识别。

yum install ntfs-3g

如果是fat32的，那么可以直接进行挂载。

bash

mkdir /home/udisk # 此处创建目录udisk并不是非要这个名称，是因为想做到见名知意。

lsblk # 用来查看文件系统

mount -o iocharset=utf8 /dev/sdb1 /home/udisk

上面的命令，其中 "-o iocharset=utf8" 是用于解决U盘挂载之后的字符乱码问题。

取消挂载：

umount 挂载点

可以通过df -h的命令查看当前文件系统的状态。如下图：

```
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2        18G  3.5G   14G  21% /
tmpfs           491M     0  491M   0% /dev/shm
/dev/sda1       194M   30M  155M  16% /boot
/dev/sr0        4.2G  4.2G     0 100% /home/cdrom
/dev/sdb4       400M  299M  102M  75% /home/udisk
```

umount /home/udisk

无法取消挂载：如图

```
[root@localhost cdrom]# umount /home/cdrom/
umount: /home/cdrom: device is busy.
        (In some cases useful info about processes that use
         the device is found by lsof(8) or fuser(1))
```

图中出现的情况是由于当前root账户处于cdrom目录中，所以导致无法取消挂载。还有其他可能是由于别的用户或软件仍在使用该目录中的文件所导致。

网络相关设备

9:41

OSI七层网络模型	TCP/IP四层概念模型	对应网络协议
应用层（Application）		HTTP、TFTP, FTP, NFS, WAIS、
表示层（Presentation）	应用层	Telnet, Rlogin, SNMP, Gopher
会话层（Session）		SMTP, DNS
传输层（Transport）	传输层	TCP, UDP
网络层（Network）	网络层	IP, ICMP, ARP, RARP, AKP, UUCP
数据链路层（Data Link）	数据链路层	FDDI, Ethernet, Arpanet, PDN, SLIP, PPP
物理层（Physical）		IEEE 802.1A, IEEE 802.2到IEEE 802.11

OSI七层和TCP/IP四层的关系

1 OSI引入了服务、接口、协议、分层的概念，TCP/IP借鉴了OSI的这些概念建立TCP/IP模型。

2 OSI先有模型，后有协议，先有标准，后进行实践；而TCP/IP则相反，先有协议和应用再提出了模型，且是参照的OSI模型。

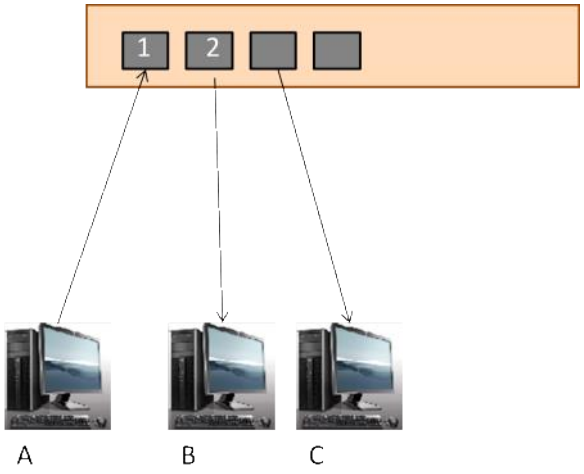
3 OSI是一种理论下的模型，而TCP/IP已被广泛使用，成为网络互联事实上的标准。

TCP：transmission control protocol 传输控制协议（打电话）

UDP：user data protocol 用户数据包协议（相当于写信）

交换机：

由来，早期的时候并没有这个设备，当时使用HUB设备进行数据的发送，但是HUB发送数据的特性广播，这样的方式，数据不安全，增加网路带宽的消耗。



交换的工作原理，通电之后，在自己的内部建立一张设备Mac地址表。这张表中记录了设备的Mac地址或其他的信息。

交换机可以组建局域网(内网)。

路由器：

(实现网络代理的功能，在公共网络上，上网的节点是路由器，并不是电脑。)

将局域网中的数据转发至公共网络(外网)。

早期上网的方式是，通过电话线插入电脑上，进行拨号上网。此方式的缺点，1、电话一直占线。2、有可能电话进来，掉线。

通信公司发现这样的缺点之后，出现了一个新的设备，这个设备的功能可以将电话线路一分为二。分别实现电话、上网的功能。调制解调器(猫)。

随着社会的发展，发现这种方式又不能满足日常的上网需求。一个猫只有一个接口，无法满足多台设备上网的需求的。

将多台设备接入交换，交换机的其中一个接口连接路由器。

连接顺序：猫上出的网线接入路由器的WAN口，路由器的LAN口接入交换机。

无线设备。大约在2008年，家用无线路由器开始普及。

无线路由器是将交换机和有线路由器和在一起产品。

网络地址相关

10:26

配置网络的时候需要配置哪些信息：

IP地址	PC在网络中的通信地址。
子网掩码	子网掩码有且只有这一个功能，用于划分网络，将一个IP地址中的网络位和主机位进行划分。是一个32位的地址。32位/24位/16位/8位
网关	网络的关口，用于数据转发，通常理解为路由器的地址，大部分硬件厂家的出厂默认地址是，192.168.0.1 | 192.168.1.1
DNS	用于解析域名的作用，Domain Name System 域名解析系统。

IP地址分析

IP地址=网络位+主机位

相同的网络，网络位肯定相同，主机位不一样

不同的网络，网络位肯定不同，主机位可能一样

比如：电话号码

网络　主机

北京：010-88889999

上海：021-12345678

021-88889999

在网络中，一般来说.0这个IP被用来当作网段的标识。255这个IP被用来当作广播地址使用，正常使用的IP范围中，其中一个IP地址要被拿来当作网关(路由器)使用。

一个网络中有多少个IP地址，取决于子网掩码。

比如：家用网络中，一般都是

192.168.1.2-254

255.255.255.0

192.168.1.1

其中，192.168.1.2-254为主机IP地址。255.255.255.0子网掩码(用于划分网络，子网掩码可以计算IP地址的数量。)。192.168.1.1作为网关使用。

192.168.1.0用来表示网段。

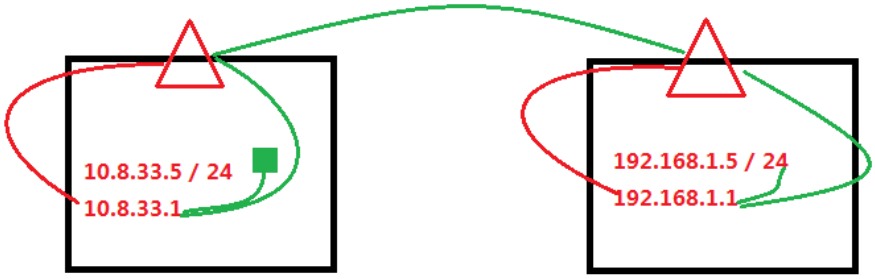
案例：

公司建立机房，决定投资建设10000台服务器的机房，那么设计网络时，掩码应该如何设计，网关应该如何设计。

16位掩码有效IP地址65534个。完全可以满足10000台服务器的需求。

网关建议大家紧贴广播地址。当前这个网络中，网关地址是192.168.255.254。

网络传输过程：



DNS：

静态：

优点	可以使我们PC/服务器有一个更快的解析速度。维护方式是手动配置服务器上hosts文件。
缺点	hosts一般都是为本机系统所有，维护一台服务器还好说。如果是上千台集群，那么维护的工作很困难

动态：

优点	只需要给服务器指明DNS服务器地址即可，无需手动配置hosts文件
缺点	有一定响应时间，(延迟)。若DNS服务器宕机，那么我们就立即失去访问域名的能力。

NAT和桥接的优缺点

11:18

桥接：

优点：	同一个局域网中的任意一台物理机想要访问虚拟机时，只要拥有账户和密码，就可以直接进行通信。
缺点：	如果宿主主机没有连接网络，那么虚拟机也就不存在与该真实网络环境中，换句话，虚拟机使用桥接模式的时候，它的网络依赖于宿主的网络环境。

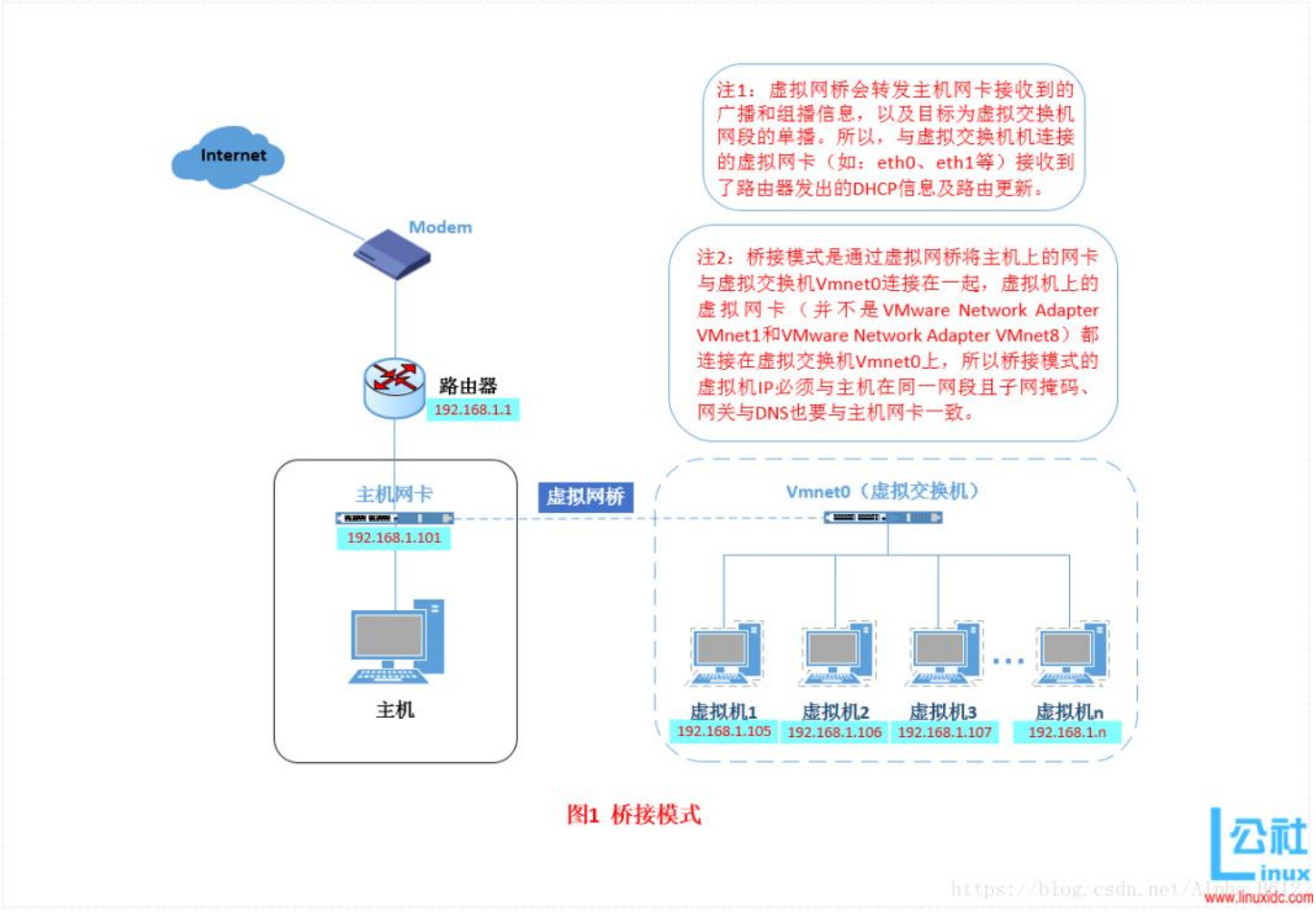


图1 桥接模式

NAT：

优点：	可以无视物理机(宿主主机)网络环境。即便是物理机没有网络，也不影响本机和虚拟机进行通信，也不影响本机上的其他虚拟机之间互相通信。因为虚拟机真正通信网卡是VMNet8提供(网络环境)
缺点：	其他物理机想要访问NAT模式下的虚拟机时，比较麻烦。

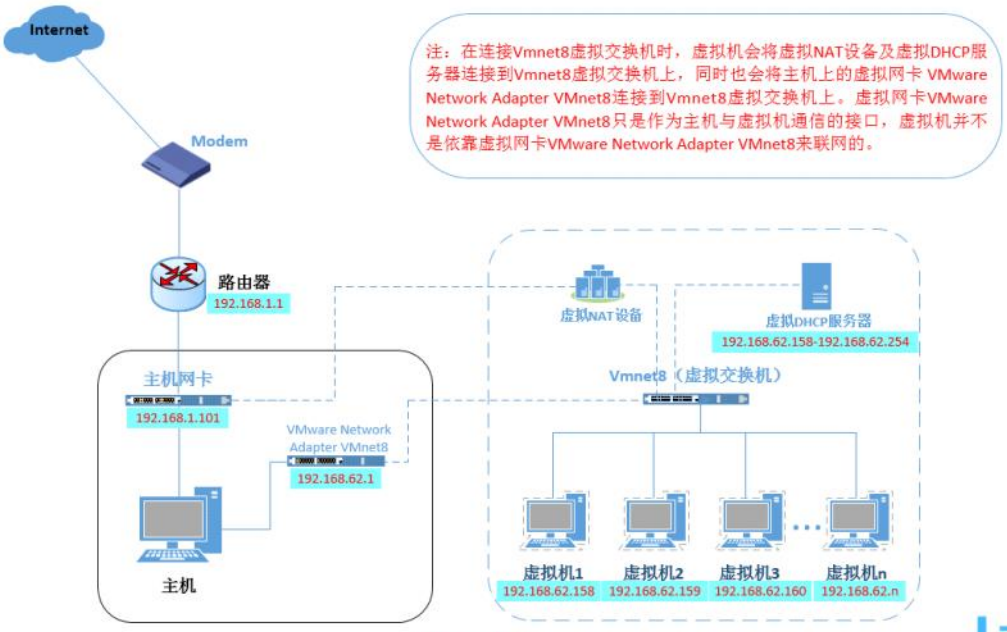


图2 NAT模式