牛犇

学习资料链接地址：

正式课程笔记：码云

https://gitee.com/niubenlinux/nsd2011

下载今日所需软件：MobaXterm\_Professinal\_20.3\_Preview4.zip

前面习题：参考答案

案例1：复制、粘贴、移动

以root用户新建/example/目录，在此目录下新建nsd.txt文件，并进一步完成下列操作

1）将“I love Study”写入到文件nsd.txt

2）将nsd.txt重命名为mylove.txt

3）将/etc/passwd、/boot、/etc/group同时拷贝到/example/目录下

4）将ifconfig命令的前两行内容，追加写入mylove.txt

5）将主机名永久配置文件，拷贝到/example/目录下

6）将DNS永久配置文件，拷贝到/example/目录下

7）将开机自动挂载配置文件，拷贝到/example/目录下

[root@A ~]# mkdir /example

[root@A ~]# echo I love Study > /example/nsd.txt

[root@A ~]# cat /example/nsd.txt

[root@A ~]# mv /example/nsd.txt /example/mylove.txt

[root@A ~]# cp -r /etc/passwd /boot /etc/group /example

[root@A ~]# ls /example

[root@A ~]# ifconfig | head -2

[root@A ~]# ifconfig | head -2 >> /example/mylove.txt

[root@A ~]# cp /etc/hostname /example

[root@A ~]# ls /example

[root@A ~]# cp /etc/resolv.conf /example

[root@A ~]# ls /example

[root@A ~]# cp /etc/fstab /example

[root@A ~]# ls /example

案例2:虚拟机上操作,（MBR分区模式）规划分区

添加一块60G的硬盘并规划分区：

划分2个10G的主分区；1个12G的主分区;1个20G的逻辑分区。

[root@A ~]# lsblk

[root@A ~]# fdisk /dev/sdd

n 创建新的分区----->分区类型 回车----->分区编号 回车---->起始扇区 回车----->在last结束时 +10G

p 查看分区表

n 创建新的分区----->分区类型 回车----->分区编号 回车---->起始扇区 回车----->在last结束时 +10G

p 查看分区表

n 创建新的分区----->分区类型 回车----->分区编号 回车---->起始扇区 回车----->在last结束时 +12G

p 查看分区表

n 创建扩展分区 ----->回车---->起始回车----->结束回车 将所有空间给扩展分区

p 查看分区表

n 创建逻辑分区----->起始回车------>结束+20G

p 查看分区表

w 保存并退出

[root@A ~]# partprobe

[root@A ~]# lsblk

[root@A ~]# ls /dev/sdd[1-5]

案例3:虚拟机上操作,分区使用

1、案例2中新添加60G硬盘的第一个逻辑分区

– 格式化成xfs文件系统，实现该分区开机自动挂载，挂载点为/mnt/xfs

2、案例2中新添加60G硬盘的第一个主分区

– 完成开机自动挂载，挂载点/mnt/mypart,文件系统为ext4

[root@A ~]# mkdir /mnt/xfs

[root@A ~]# mkfs.xfs /dev/sdd5

[root@A ~]# vim /etc/fstab

/dev/sdd5 /mnt/xfs xfs defaults 0 0

[root@A ~]# mount -a

[root@A ~]# df -h

[root@A ~]# mkdir /mnt/mypart

[root@A ~]# mkfs.ext4 /dev/sdd1

[root@A ~]# vim /etc/fstab

/dev/sdd1 /mnt/mypart ext4 defaults 0 0

[root@A ~]# mount -a

[root@A ~]# df -h

案例4:虚拟机上操作,采用GPT分区模式，利用parted规划分区

添加一块20G的硬盘并规划分区：

划分2个2G的主分区；1个5G的主分区;

[root@A ~]# lsblk

[root@A ~]# parted /dev/sde

(parted) mktable gpt

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 0

结束点？ 2G

警告: The resulting partition is not properly aligned for best performance.

忽略/Ignore/放弃/Cancel? Ignore #选择忽略，给出存放分区表信息的空间

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 2G

结束点？ 4G

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 4G

结束点？ 9G

(parted) print

(parted) quit

[root@A ~]# lsblk

[root@A ~]# ls /dev/sde[1-3]

案例5:虚拟机上操作,交换分区使用

1、案例4中新添加20G硬盘的第一个主分区

– 格式化成ext4文件系统，实现该分区开机挂载到/stu01

2、案例4中新添加20G硬盘的第二个主分区

– 格式化成xfs文件系统，实现该分区开机挂载到/stu02

[root@A ~]# mkdir /stu01

[root@A ~]# mkfs.ext4 /dev/sde1

[root@A ~]# vim /etc/fstab

/dev/sde1 /stu01 ext4 defaults 0 0

[root@A ~]# mount -a

[root@A ~]# df -h

[root@A ~]# mkdir /stu02

[root@A ~]# mkfs.xfs /dev/sde2

[root@A ~]# vim /etc/fstab

/dev/sde2 /stu02 xfs defaults 0 0

[root@A ~]# mount -a

[root@A ~]# df -h

案例6:虚拟机上操作,交换分区使用

1、案例4中新添加20G硬盘的第三个主分区

– 格式化成交换文件系统，实现该分区开机自动启用

[root@A ~]# swapon -s

[root@A ~]# mkswap /dev/sde3

[root@A ~]# blkid /dev/sde3

[root@A ~]# vim /etc/fstab

/dev/sde3 swap swap defaults 0 0

[root@A ~]# swapon -a

[root@A ~]# swapon -s

案例7:虚拟机上操作,文件扩展Swap空间

1. 使用dd命令创建一个大小为2048MB的交换文件，放在/opt/swap.db

2. 将swap.db文件格式化成swap文件系统

3. 启用swap.db文件，查看swap空间组成

4. 停用swap.db文件，查看swap空间组成

[root@A ~]# dd if=/dev/zero of=/opt/swap.db bs=1M count=2048

[root@A ~]# ls -lh /opt/swap.db

[root@A ~]# mkswap -f /opt/swap.db #-f强制格式化（可以不添加）

[root@A ~]# blkid /opt/swap.db

[root@A ~]# swapon /opt/swap.db

[root@A ~]# swapon -s

[root@A ~]# swapoff /opt/swap.db

[root@A ~]# swapon -s

前面习题：参考答案

案例1：复制、粘贴、移动

以root用户新建/exam/目录，在此目录下新建king.txt文件，并进一步完成下列操作

1）将“I Love hehe”写入到文件king.txt

2）将king.txt重命名为my.txt

3）将/etc/passwd、/boot、/etc/group同时拷贝到/exam/目录下

4）将ifconfig命令的前两行内容，追加写入king.txt

5）将主机名永久配置文件，拷贝到/exam/目录下

6）将存放组基本信息的配置文件，拷贝到/exam/目录下

7）将开机自动挂载配置文件，拷贝到/exam/目录下

[root@A ~]# mkdir /exam

[root@A ~]# echo I Love hehe > /exam/king.txt

[root@A ~]# cat /exam/king.txt

[root@A ~]# mv /exam/king.txt /exam/my.txt

[root@A ~]# cp -r /etc/passwd /boot /etc/group /exam

[root@A ~]# ifconfig | head -2

[root@A ~]# ifconfig | head -2 >> /exam/king.txt

[root@A ~]# cp /etc/hostname /exam/

[root@A ~]# ls /exam

[root@A ~]# cp /etc/group /exam

[root@A ~]# ls /exam

[root@A ~]# cp /etc/fstab /exam

[root@A ~]# ls /exam

案例2:虚拟机上操作,采用GPT分区模式，利用parted规划分区

添加一块30G的硬盘并规划分区：

划分2个2G的主分区；1个5G的主分区;

[root@A ~]# parted /dev/sdb

(parted) mktable gpt

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 0

结束点？ 2G

警告: The resulting partition is not properly aligned for best performance.

忽略/Ignore/放弃/Cancel? Ignore #选择忽略，给出存放分区表信息的空间

(parted) print

(parted) unit GB #使用GB作为显示的单位

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 2G

结束点？ 4G

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 4G

结束点？ 8G

(parted) print

(parted) quit

[root@A ~]# lsblk

案例3:虚拟机上操作,交换分区使用

1、案例2中新添加30G硬盘的第一个主分区

– 格式化成交换文件系统，实现该分区开机自动启用

2、案例2中新添加30G硬盘的第二个主分区

– 格式化成交换文件系统，实现该分区开机自动启用

[root@A ~]# mkswap /dev/sdb1

[root@A ~]# blkid /dev/sdb1

[root@A ~]# swapon

[root@A ~]# swapon /dev/sdb1

[root@A ~]# swapon

[root@A ~]# free -m

[root@A ~]# mkswap /dev/sdb2

[root@A ~]# blkid /dev/sdb2

[root@A ~]# swapon

[root@A ~]# swapon /dev/sdb2

[root@A ~]# swapon

[root@A ~]# free -m

案例4:虚拟机上操作,文件扩展Swap空间

1. 使用dd命令创建一个大小为2048MB的交换文件，放在/opt/swap.db

2. 将swap.db文件格式化成swap文件系统

3. 启用swap.db文件，查看swap空间组成

4. 停用swap.db文件，查看swap空间组成

[root@A ~]# dd if=/dev/zero of=/opt/swap.db bs=1M count=2048

[root@A ~]# ls -lh /opt/swap.db

[root@A ~]# mkswap -f /opt/swap.db

[root@A ~]# blkid /opt/swap.db

[root@A ~]# swapon

[root@A ~]# swapon /opt/swap.db

[root@A ~]# swapon

[root@A ~]# free -m

[root@A ~]# swapoff /opt/swap.db

[root@A ~]# swapon

[root@A ~]# free -m

案例5:虚拟机操作，构建 LVM 存储

– 新建一个名为 systemvg 的卷组

– 在此卷组中创建一个名为 vo 的逻辑卷，大小为8G

– 将逻辑卷 vo 格式化为 xfs 文件系统

– 将逻辑卷 vo 挂载到 /vo 目录，并在此目录下建立一个测试文件 votest.txt，内容为“I AM KING.”

– 实现逻辑卷vo开机自动挂载到/vo

[root@A ~]# parted /dev/sdb

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 8G

结束点？ 100%

(parted) print

(parted) quit

[root@A ~]# vgcreate systemvg /dev/sdb4

[root@A ~]# vgs

[root@A ~]# lvcreate -n vo -L 8G systemvg

[root@A ~]# lvs

[root@A ~]# mkfs.xfs /dev/systemvg/vo

[root@A ~]# blkid /dev/systemvg/vo

[root@A ~]# vim /etc/fstab

/dev/systemvg/vo /vo xfs defaults 0 0

[root@A ~]# mkdir /vo

[root@A ~]# mount -a

[root@A ~]# df -h

[root@A ~]# echo I AM KING > /vo/votest.txt

案例6:虚拟机操作，构建 LVM 存储(修改PE大小)

– 新的逻辑卷命名为 database，其大小为50个PE的大小，属于 datastore 卷组

– 在 datastore 卷组中其PE的大小为1M

– 使用 EXT4 文件系统对逻辑卷 database 格式化，此逻辑卷应该在开机时自动挂载到 /mnt/database 目录

[root@A ~]# vgcreate -s 1M datastore /dev/sdb3

[root@A ~]# vgs

[root@A ~]# vgdisplay datastore

[root@A ~]# lvcreate -n database -l 50 datastore

[root@A ~]# lvs

[root@A ~]# mkfs.ext4 /dev/datastore/database

[root@A ~]# blkid /dev/datastore/database

[root@A ~]# vim /etc/fstab

/dev/datastore/database /mnt/database ext4 defaults 0 0

[root@A ~]# mkdir /mnt/database

[root@A ~]# mount -a

[root@A ~]# df -h

案例7:虚拟机操作，扩展逻辑卷

– 将/dev/systemvg/vo逻辑卷的大小扩展到20G

[root@A ~]# vgs

[root@A ~]# lvextend -L 20G /dev/systemvg/vo

[root@A ~]# vgs

[root@A ~]# lvs

[root@A ~]# df -h

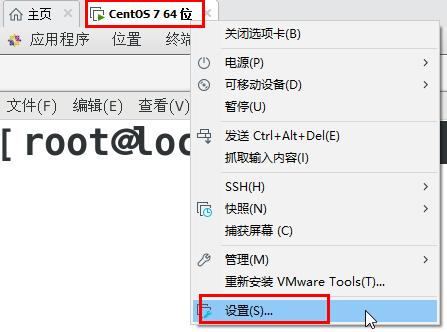
[root@A ~]# xfs\_growfs /dev/systemvg/vo

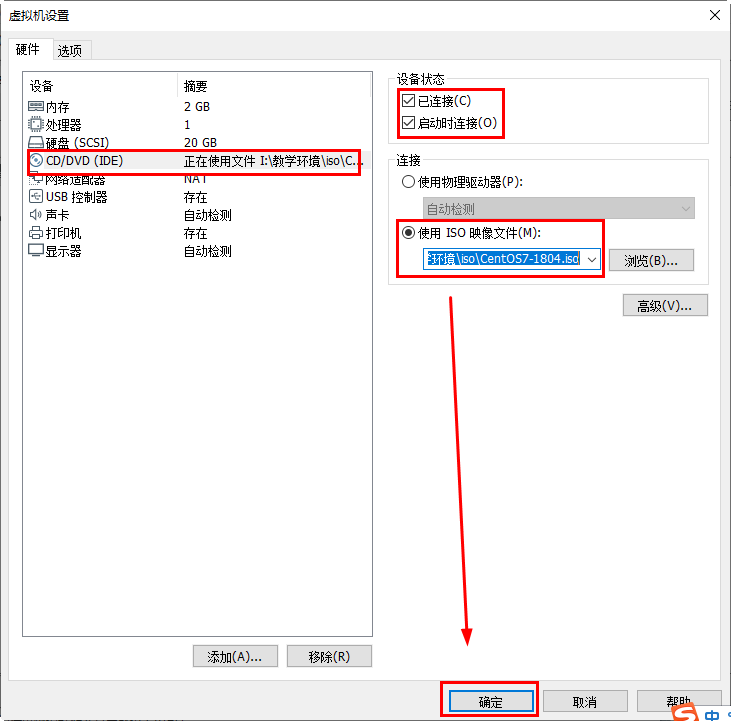
[root@A ~]# df -h

####################################################

1. 构建Yum仓库，环境准备

1.光盘文件放入光驱设备





2.挂载光驱设备

[root@localhost ~]# mkdir /mydvd

[root@localhost ~]# mount /dev/cdrom /mydvd [root@localhost ~]# ls /mydvd

3.书写客户端配置文件

[root@localhost ~]# rm -rf /etc/yum.repos.d/\*

[root@localhost ~]# vim /etc/yum.repos.d/mydvd.repo

[centos]

name=2011

baseurl=file:///mydvd

enabled=1

gpgcheck=0

[root@localhost ~]# yum -y install xorg-x11-apps

[root@localhost ~]# rpm -q xorg-x11-apps

4．完成开机自动挂载

]# blkid /dev/cdrom #查看 光驱设备文件系统类型

]# vim /etc/fstab

/dev/cdrom /mydvd iso9660 defaults 0 0

]# umount /mydvd

]# mount -a

]# ls /mydvd

]# reboot #重启测试

]# yum -y install httpd #测试安装软件包

1. 配置网络参数之主机名

* **配置永久的主机名**

]# echo svr7.tedu.cn > /etc/hostname

]# cat /etc/hostname

svr7.tedu.cn

]# hostname svr7.tedu.cn #修改当前

]# hostname

svr7.tedu.cn

开启一个新的终端查看提示符的变化

1. 配置网络参数之IP地址与子网掩码、网关地址

* **修改网卡命令规则(eth0、eth1、eth2……)**

]# ifconfig | head -2

**ens33**: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

ether 00:0c:29:8a:72:4f txqueuelen 1000 (Ethernet)

]# **vim /etc/default/grub** #grub内核引导程序

……..此处省略一万字

GRUB\_CMDLINE\_LINUX="…….. quiet **net.ifnames=0 biosdevname=0**"

……..此处省略一万字

]# **grub2-mkconfig -o /boot/grub2/grub.cfg** #让网卡命名规则生效

]# **reboot**

]# **ifconfig | head -2**

**eth0**: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.81.132 netmask 255.255.255.0 broadcast 192.168.81.255

* **nmcli命令的网卡命名，删除错误网卡命名**

[root@svr7 ~]# **nmcli connection show** #查看

[root@svr7 ~]# **nmcli connection delete ens33**

[root@svr7 ~]# **nmcli connection show**

[root@svr7 ~]# **nmcli connection show**

[root@svr7 ~]# **nmcli connection delete 有线连接\ 1**

[root@svr7 ~]# **nmcli connection show**

* **nmcli命令的网卡命名，添加网卡命名**

[root@svr7 ~]# **nmcli connection add type ethernet ifname eth0 con-name eth0**

解析： nmcli connection 添加 类型 以太网设备

网卡设备名为eth0 nmcli命令的命名为eth0

[root@svr7 ~]# **nmcli connection show**

* **修改IP地址、子网掩码、网关地址**

[root@svr7 ~]# **nmcli connection modify eth0**

**ipv4.method manual**

**ipv4.addresses 192.168.4.7/24**

**ipv4.gateway 192.168.4.254**

**connection.autoconnect yes**

[root@svr7 ~]# nmcli connection 修改 网卡名

ipv4.方法 手工配置

ipv4.地址 192.168.4.7/24

ipv4.网关 192.168.4.254

每次开机自动启用

[root@svr7 ~]# **nmcli connection up eth0** #激活

[root@svr7 ~]# **ifconfig | head -2**

网卡配置文件：/etc/sysconfig/network-scripts/ifcfg-eth0

]# cat /etc/sysconfig/network-scripts/ifcfg-eth0

]# route -n #查看网关地址信息

Gateway

**192.168.4.254**

1. 配置网络参数之DNS服务器地址

DNS服务器：负责域名解析的机器，将域名解析为IP地址

[root@svr7 ~]# echo nameserver 8.8.8.8 > /etc/resolv.conf

[root@svr7 ~]# cat /etc/resolv.conf

nameserver 8.8.8.8

1. 模板机器的修改

将UUID进行修改，修改为/dev/sda1

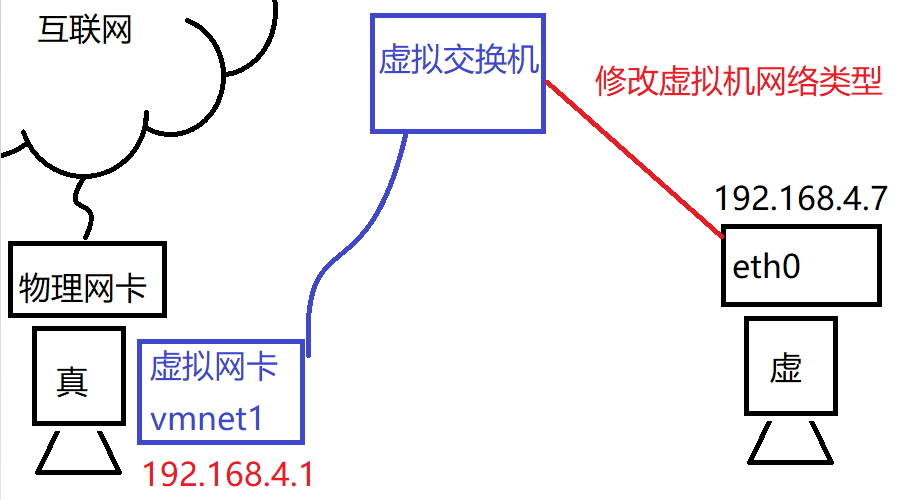
[root@svr7 ~]# vim /etc/fstab

…….此处省略一万字

**/dev/sda1**  /boot xfs defaults 0 0

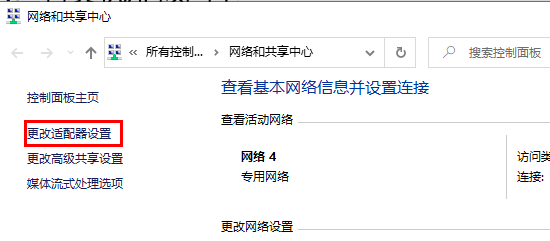
……. 此处省略一万字

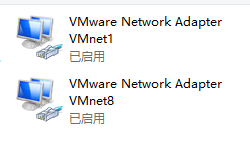
1. 真机与虚拟机的通信



1.查看真机虚拟网卡





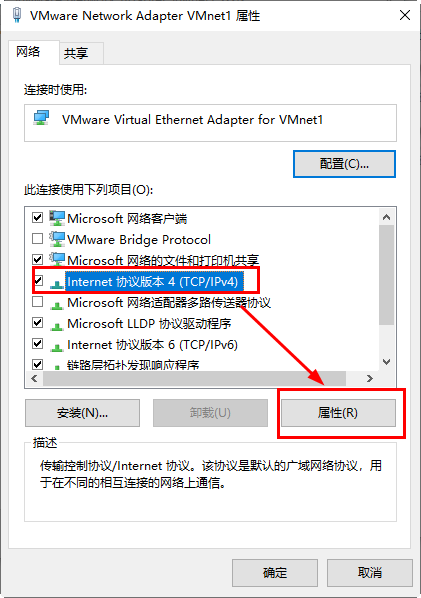


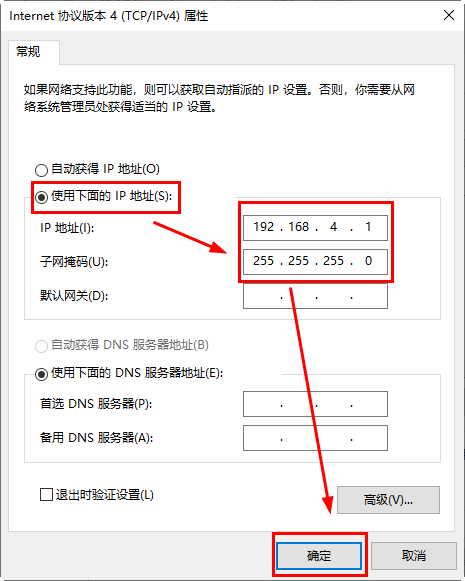
2.真机配置VMnet1的网卡IP地址为192.168.4.1

**双击**VMnet1网络适配器





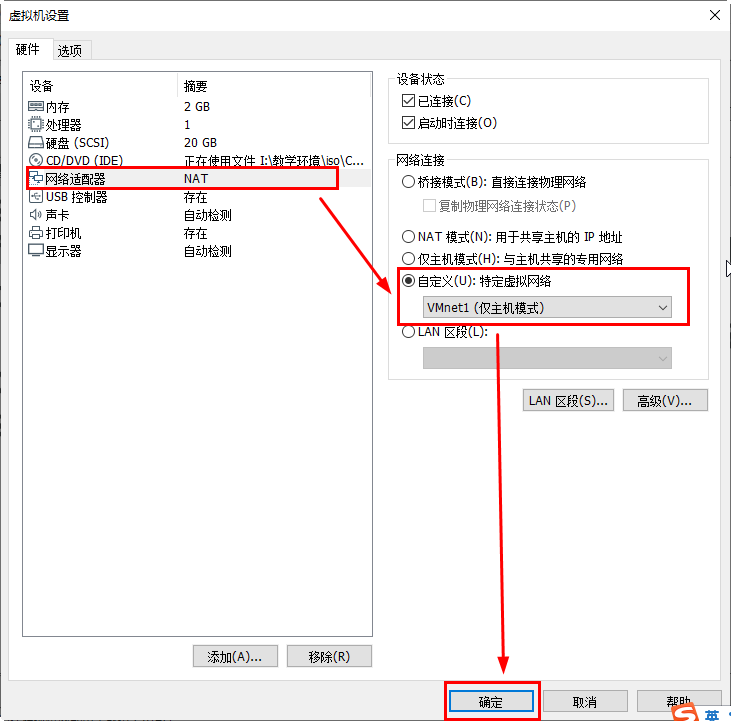




3.配置**虚拟机**网络类型

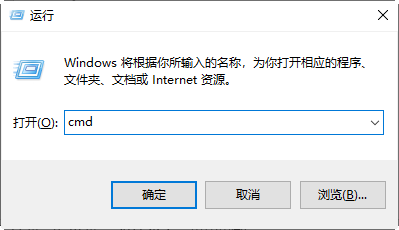
右击----》选择设置

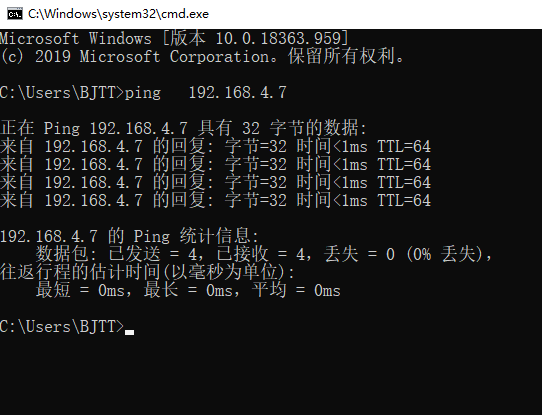




4.测试通信

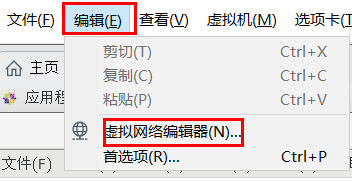
Windows键+r快捷键





常见问题：

1.虚拟网卡vmnet1没有



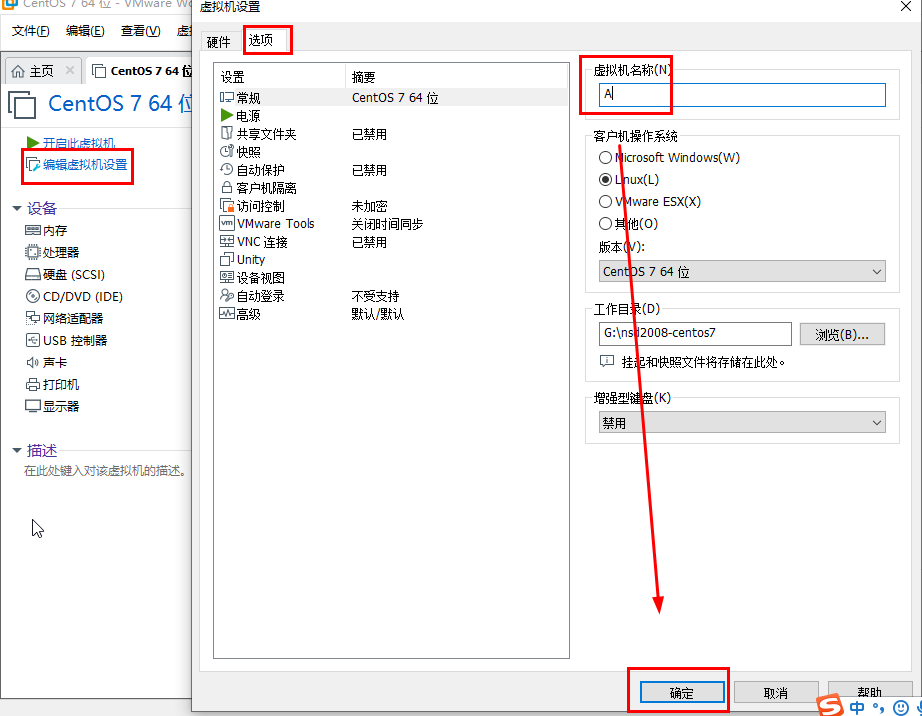




如果还是无法出现vmnet1，参考下列网站方法

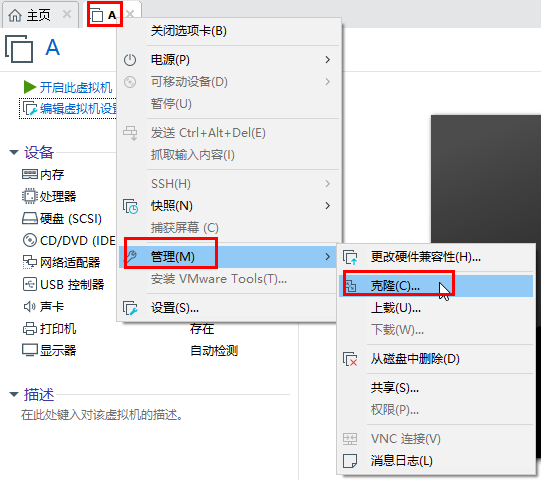
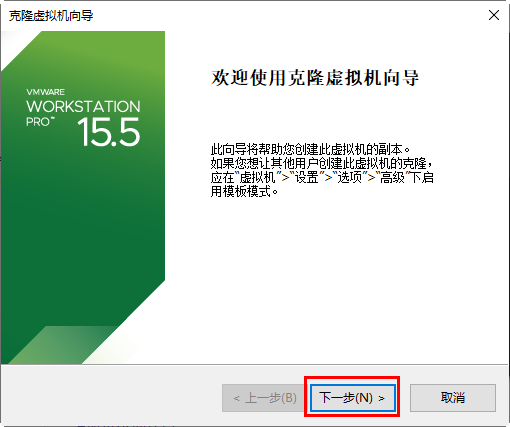
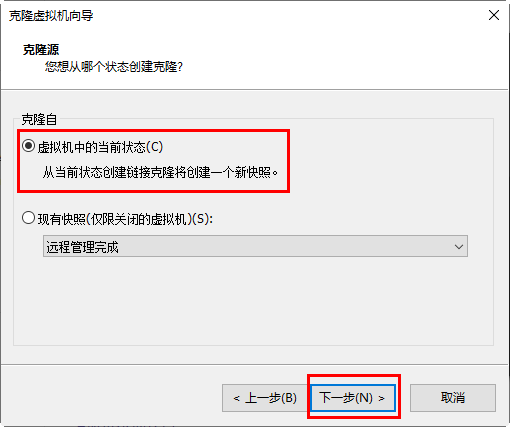
<https://jingyan.baidu.com/article/066074d6f19bd0c3c31cb048.html>

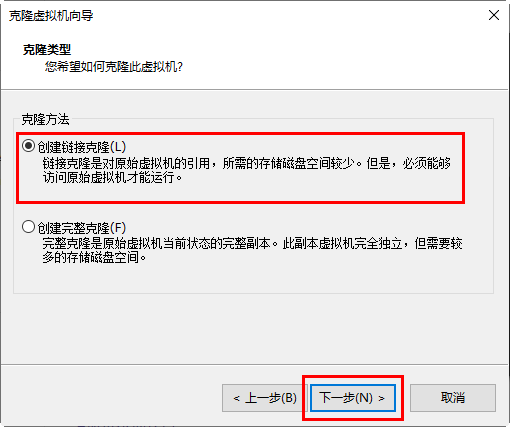
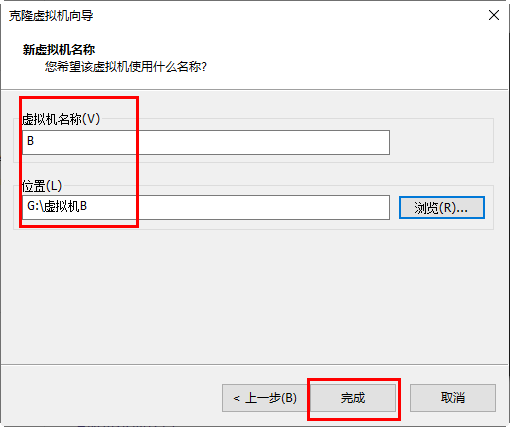
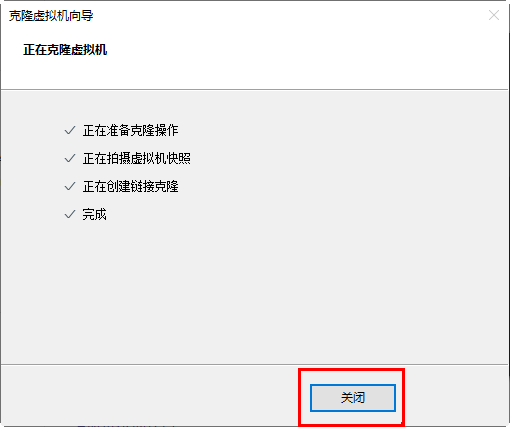
1. 虚拟机拍摄快照

[root@svr7 ~]# poweroff



1. 克隆虚拟机A（克隆必须关闭模板机器）





1. 虚拟机B配置

虚拟机B：

[root@svr7 ~]# hostname pc207.tedu.cn

[root@svr7 ~]# echo pc207.tedu.cn > /etc/hostname

[root@svr7 ~]# hostname

pc207.tedu.cn

新开一个终端进行测试

[root@pc207 ~]# nmcli connection modify eth0 ipv4.method manual ipv4.addresses **192.168.4.207/24** connection.autoconnect yes

[root@pc207 ~]# nmcli connection up eth0

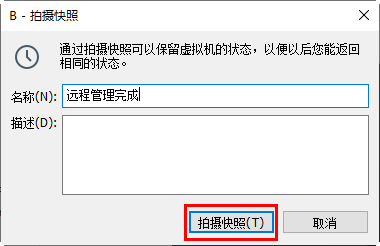
[root@pc207 ~]# ifconfig | head -2

[root@pc207 ~]# ping 192.168.4.7

1. 虚拟机B拍摄快照

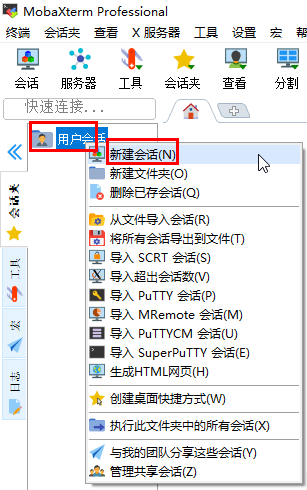
[root@pc207 ~]# poweroff





1. 利用真机windows进行远程管理







Ctrl+滚轮=可以放大或变小字体

1. 远程管理(Linux与Linux)

* **软件包的安装**

[root@svr7 /]# rpm -qa | grep openssh

openssh-7.4p1-16.el7.x86\_64

openssh-server-7.4p1-16.el7.x86\_64

openssh-clients-7.4p1-16.el7.x86\_64

* **远程登录工具 ssh**

**虚拟机A：**

[root@svr7 /]# **ssh root@192.168.4.207**

………necting (yes/no)? **yes**

root@192.168.4.207's password:  **#输入密码**

[root@pc207 ~]# **touch /root/hahaxixi.txt**

[root@pc207 ~]# **exit #退出**

登出

Connection to 192.168.4.207 closed.

[root@svr7 /]# cat /root/.ssh/known\_hosts #记录曾经远程管理的机器

* **实现ssh远程管理无密码验证**

虚拟机A：

1.生成公钥(锁)与私钥(钥匙)进行验证

[root@svr7 ~]# **ssh-keygen**  #一路回车

…….save the key (/root/.ssh/id\_rsa): #回车 设置默认保存位置

……..assphrase): #回车 设置密码为空

…….. again: #回车 设置密码为空

[root@svr7 ~]# ls /root/.ssh/

id\_rsa(私钥) id\_rsa.pub(公钥) known\_hosts

2.将公钥(锁)传递给虚拟机B

[root@svr7 ~]# **ssh-copy-id root@192.168.4.207**

[root@svr7 ~]# ssh root@192.168.4.207 #测试无密码

[root@pc207 ~]# exit

登出

Connection to 192.168.4.207 closed.

[root@svr7 ~]#

虚拟机B

[root@pc207 ~]# ls /root/.ssh/

authorized\_keys(别的机器传递过来的公钥) known\_hosts

[root@pc207 ~]#

* **安全复制工具 scp=ssh+cp**
  + scp [-r] 用户名@服务器:路径 本地路径
  + scp [-r] 本地路径 用户名@服务器:路径

虚拟机A：

]# scp /etc/passwd root@192.168.4.207:/root

]# scp -r /home root@192.168.4.207:/root/

]# scp root@192.168.4.207:/etc/shadow /mnt/

虚拟机B：

]# ls /root

1. **自定义**Yum仓库

将自己下载的众多RPM软件，构建成Yum仓库

服务端：完美的仓库（1.众多的软件包 **2.仓库数据文件**）

* **将真机的tools.tar.gz传递数据到虚拟机A**



[root@svr7 ~]# ls /root

**tools.tar.gz**  下载

公共 音乐

[root@svr7 ~]#

* **解包**

[root@svr7 ~]# tar -tf /root/tools.tar.gz #查看

[root@svr7 ~]# tar -xf /root/tools.tar.gz -C / #解包

[root@svr7 ~]# ls /

[root@svr7 ~]# ls /tools

[root@svr7 ~]# ls /tools/other

* **建立仓库数据文件**

[root@svr7 ~]# createrepo /tools/other/

[root@svr7 ~]# ls /tools/other

* **书写客户端配置文件**

[root@svr7 ~]# vim /etc/yum.repos.d/mydvd.repo

[centos]

name=2011

baseurl=file:///mydvd

enabled=1

gpgcheck=0

**[myrpm]**

name=2011

baseurl=**file:///tools/other**

enabled=1

gpgcheck=0

[root@svr7 ~]# yum repolist

[root@svr7 ~]# yum -y install sl

[root@svr7 ~]# sl

[root@svr7 ~]# yum -y install cmatrix

[root@svr7 ~]# cmatrix

1. 源码编译安装

RPM包：rpm -ivh yum install

源码包---开发工具--->可以执行的程序------>运行安装

* 主要优点
  + 获得软件的最新版，及时修复bug
  + 软件功能可按需选择/定制，有更多软件可供选择
  + 源码包适用各种平台
  + ……
* **将真机的tools.tar.gz传递数据到虚拟机A**



[root@svr7 ~]# ls /root

**tools.tar.gz**  下载

公共 音乐

[root@svr7 ~]#

源码编译安装步骤：

步骤一:安装开发工具

]# yum -y install make

]# yum -y install gcc

]# rpm -q gcc

]# rpm -q make

步骤二:进行tar解包

]# tar -xf /root/tools.tar.gz -C /usr/local/

]# ls /usr/local/

]# ls /usr/local/tools/

]# tar -xf /usr/local/tools/inotify-tools-3.13.tar.gz -C /usr/local/

]# ls /usr/local/

步骤三：运行configure脚本进行配置

作用1：检测系统是否安装gcc

作用2：可以指定安装位置及功能

]# cd /usr/local/inotify-tools-3.13/

]# ./configure --prefix=/opt/myrpm #指定安装位置

常见错误：没有安装gcc

checking for gcc... no

checking for cc... no

checking for cl.exe... no

configure: error: no acceptable C compiler found in $PATH

See `config.log' for more details.

步骤四：make进行编译，产生可以执行的程序

]# cd /usr/local/inotify-tools-3.13/

]# make

步骤五：make install进行安装

]# cd /usr/local/inotify-tools-3.13/

]# make install

]# ls /opt/

]# ls /opt/myrpm/

]# ls /opt/myrpm/bin/

课后习题：

案例1:虚拟机B上操作：实现静态网络参数配置

– 主机名:test.example.com

– IP地址:172.25.0.11

– 子网掩码:255.255.0.0

– 默认网关:172.25.0.254

– DNS服务器:172.25.254.254

案例2:虚拟机B上操作：实现静态网络参数配置

– 主机名:B.tedu.cn

– IP地址:192.168.1.1

– 子网掩码:255.255.255.0

– 默认网关:192.168.1.254

– DNS服务器:8.8.8.8

案例3:虚拟机B上操作：实现静态网络参数配置

– 主机名:pc207.tedu.cn

– IP地址:192.168.4.207

– 子网掩码:255.255.255.0

– 默认网关:192.168.4.254

– DNS服务器:1.1.1.1

案例4:虚拟机A上操作：实现静态网络参数配置

– 主机名:svr7.tedu.cn

– IP地址:192.168.4.7

– 子网掩码:255.255.255.0

– 默认网关:192.168.4.254

– DNS服务器:1.1.1.1

案例5：虚拟机B：传递数据

1.将本机/usr/local/tools/other目录传递到虚拟机A，放在虚拟机A的/usr/目录下

2.将本机/etc/gshadow文件传递到虚拟机A，放在虚拟机A的/root目录下

3.将本机/etc/skel目录传递到虚拟机A，放在虚拟机A的/tmp目录下

案例6：虚拟机B：远程无密码验证

1.实现虚拟机B远程管理虚拟机A，无需密码验证

昨日习题：

案例1：复制、粘贴、移动

以root用户新建/exam/目录，在此目录下新建king.txt文件，并进一步完成下列操作

1）将“I Love hehe”写入到文件king.txt

2）将king.txt重命名为my.txt

3）将/etc/passwd、/boot、/etc/group同时拷贝到/exam/目录下

4）将ifconfig命令的前两行内容，追加写入king.txt

5）将主机名永久配置文件，拷贝到/exam/目录下

6）将存放组基本信息的配置文件，拷贝到/exam/目录下

7）将开机自动挂载配置文件，拷贝到/exam/目录下

[root@A ~]# mkdir /exam

[root@A ~]# echo I Love hehe > /exam/king.txt

[root@A ~]# cat /exam/king.txt

[root@A ~]# mv /exam/king.txt /exam/my.txt

[root@A ~]# cp -r /etc/passwd /boot /etc/group /exam

[root@A ~]# ifconfig | head -2

[root@A ~]# ifconfig | head -2 >> /exam/king.txt

[root@A ~]# cp /etc/hostname /exam/

[root@A ~]# ls /exam

[root@A ~]# cp /etc/group /exam

[root@A ~]# ls /exam

[root@A ~]# cp /etc/fstab /exam

[root@A ~]# ls /exam

案例2:虚拟机上操作,采用GPT分区模式，利用parted规划分区

添加一块30G的硬盘并规划分区：

划分2个2G的主分区；1个5G的主分区;

[root@A ~]# parted /dev/sdb

(parted) mktable gpt

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 0

结束点？ 2G

警告: The resulting partition is not properly aligned for best performance.

忽略/Ignore/放弃/Cancel? Ignore #选择忽略，给出存放分区表信息的空间

(parted) print

(parted) unit GB #使用GB作为显示的单位

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 2G

结束点？ 4G

(parted) print

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 4G

结束点？ 8G

(parted) print

(parted) quit

[root@A ~]# lsblk

案例3:虚拟机上操作,交换分区使用

1、案例2中新添加30G硬盘的第一个主分区

– 格式化成交换文件系统，实现该分区开机自动启用

2、案例2中新添加30G硬盘的第二个主分区

– 格式化成交换文件系统，实现该分区开机自动启用

[root@A ~]# mkswap /dev/sdb1

[root@A ~]# blkid /dev/sdb1

[root@A ~]# swapon

[root@A ~]# swapon /dev/sdb1

[root@A ~]# swapon

[root@A ~]# free -m

[root@A ~]# mkswap /dev/sdb2

[root@A ~]# blkid /dev/sdb2

[root@A ~]# swapon

[root@A ~]# swapon /dev/sdb2

[root@A ~]# swapon

[root@A ~]# free -m

案例4:虚拟机上操作,文件扩展Swap空间

1. 使用dd命令创建一个大小为2048MB的交换文件，放在/opt/swap.db

2. 将swap.db文件格式化成swap文件系统

3. 启用swap.db文件，查看swap空间组成

4. 停用swap.db文件，查看swap空间组成

[root@A ~]# dd if=/dev/zero of=/opt/swap.db bs=1M count=2048

[root@A ~]# ls -lh /opt/swap.db

[root@A ~]# mkswap -f /opt/swap.db

[root@A ~]# blkid /opt/swap.db

[root@A ~]# swapon

[root@A ~]# swapon /opt/swap.db

[root@A ~]# swapon

[root@A ~]# free -m

[root@A ~]# swapoff /opt/swap.db

[root@A ~]# swapon

[root@A ~]# free -m

案例5:虚拟机操作，构建 LVM 存储

– 新建一个名为 systemvg 的卷组

– 在此卷组中创建一个名为 vo 的逻辑卷，大小为8G

– 将逻辑卷 vo 格式化为 xfs 文件系统

– 将逻辑卷 vo 挂载到 /vo 目录，并在此目录下建立一个测试文件 votest.txt，内容为“I AM KING.”

– 实现逻辑卷vo开机自动挂载到/vo

[root@A ~]# parted /dev/sdb

(parted) mkpart

分区名称？ []? xixi

文件系统类型？ [ext2]? xfs

起始点？ 8G

结束点？ 100%

(parted) print

(parted) quit

[root@A ~]# vgcreate systemvg /dev/sdb4

[root@A ~]# vgs

[root@A ~]# lvcreate -n vo -L 8G systemvg

[root@A ~]# lvs

[root@A ~]# mkfs.xfs /dev/systemvg/vo

[root@A ~]# blkid /dev/systemvg/vo

[root@A ~]# vim /etc/fstab

/dev/systemvg/vo /vo xfs defaults 0 0

[root@A ~]# mkdir /vo

[root@A ~]# mount -a

[root@A ~]# df -h

[root@A ~]# echo I AM KING > /vo/votest.txt

案例6:虚拟机操作，构建 LVM 存储(修改PE大小)

– 新的逻辑卷命名为 database，其大小为50个PE的大小，属于 datastore 卷组

– 在 datastore 卷组中其PE的大小为1M

– 使用 EXT4 文件系统对逻辑卷 database 格式化，此逻辑卷应该在开机时自动挂载到 /mnt/database 目录

[root@A ~]# vgcreate -s 1M datastore /dev/sdb3

[root@A ~]# vgs

[root@A ~]# vgdisplay datastore

[root@A ~]# lvcreate -n database -l 50 datastore

[root@A ~]# lvs

[root@A ~]# mkfs.ext4 /dev/datastore/database

[root@A ~]# blkid /dev/datastore/database

[root@A ~]# vim /etc/fstab

/dev/datastore/database /mnt/database ext4 defaults 0 0

[root@A ~]# mkdir /mnt/database

[root@A ~]# mount -a

[root@A ~]# df -h

案例7:虚拟机 操作，扩展逻辑卷

– 将/dev/systemvg/vo逻辑卷的大小扩展到20G

[root@A ~]# vgs

[root@A ~]# lvextend -L 20G /dev/systemvg/vo

[root@A ~]# vgs

[root@A ~]# lvs

[root@A ~]# df -h

[root@A ~]# xfs\_growfs /dev/systemvg/vo

[root@A ~]# df -h

案例8:进程管理

1.查看当前系统中整个进程树信息

2.利用pstree查看lisi开启的进程

3.显示当前系统正在运行的所有进程信息

4.检索当前系统中进程，进程名包含cron的PID是多少？

[root@A ~]# pstree -ap

[root@A ~]# pstree -ap lisi

[root@A ~]# ps aux

[root@A ~]# pgrep cron