牛犇

学习资料链接地址：

正式课程笔记：码云

https://gitee.com/niubenlinux/nsd2011

**一、环境准备**

还原快照网络配置完成，开启虚拟机A与虚拟机B

**二、常用的网络工具**

* **ip命令**

1.查看IP地址

]# ip address show

]# ip a s

2.添加IP地址

]# ip address add 192.168.10.1/24 dev eth0

]# ip a s

]# ip address add 192.168.20.1/24 dev eth0

]# ip a s

* **ping 命令，测网络连接**
  + 选项 -c 包个数

[root@svr7 ~]# ping -c 2 192.168.4.7

[root@svr7 ~]# ping -c 3 192.168.4.207

三、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 #检测是否安装成功

]# rpm -q make #检测是否安装成功

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/

**四、日志管理**

* 系统和程序的“日记本”
  + 记录系统、程序运行中发生的各种事件
  + 通过查看日志，了解及排除故障
  + 信息安全控制的“依据”
* 由系统服务rsyslog统一记录/管理
  + 日志消息采用文本格式
  + 主要记录事件发生的时间、主机、进程、内容
* 常见的日志文件

|  |  |
| --- | --- |
| **日志文件** | **主要用途** |
| /var/log/messages | 记录内核消息、各种服务的公共消息 |
| /var/log/dmesg | 记录系统启动过程的各种消息 |
| /var/log/cron | 记录与cron计划任务相关的消息 |
| /var/log/maillog | 记录邮件收发相关的消息 |
| /var/log/secure | 记录与访问限制相关的安全消息 |

* 通用分析工具
  + tail、tailf、less、grep等文本浏览/检索命令
  + awk、sed等格式化过滤工具

tailf：实时跟踪

[root@svr7 /]# echo 123456 > /opt/1.txt

[root@svr7 /]# tailf /opt/1.txt

* users、who、w 命令
  + 查看已登录的用户信息，详细度不同
* last、lastb 命令
  + 查看最近登录成功/失败的用户信息

[root@svr7 /]# users

[root@svr7 /]# who

pts：图形命令行终端

[root@svr7 /]# last #登录成功的用户

[root@svr7 /]# lastb #登录失败的用户

* Linux内核定义的事件紧急程度
  + **分为 0~7 共8种优先级别**
  + **其数值越小**，表示对应事件越紧急/重要

0 EMERG（紧急） 会导致主机系统不可用的情况

1 ALERT（警告） 必须马上采取措施解决的问题

2 CRIT（严重） 比较严重的情况

3 ERR（错误） 运行出现错误

4 WARNING（提醒） 可能会影响系统功能的事件

5 NOTICE（注意） 不会影响系统但值得注意

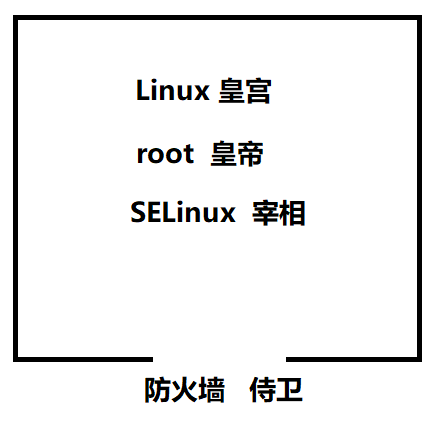
6 INFO（信息） 一般信息

7 DEBUG（调试） 程序或系统调试信息等

**五、系统安全保护**

**SELinux概述**

* Security-Enhanced Linux
  + 美国NSA国家安全局主导开发，一套增强Linux系统安全的强制访问控制体系
  + 集成到Linux内核（2.6及以上）中运行
  + RHEL7基于SELinux体系针对用户、进程、目录和文件提供了预设的保护策略，以及管理工具



* SELinux的运行模式
  + enforcing（强制）、permissive（宽松）
  + disabled（彻底禁用）

任何模式变成disabled模式，都要经历重启系统

* 切换运行模式
  + 临时切换：setenforce 1或0
  + 固定配置：/etc/selinux/config 文件

虚拟机A

[root@svr7 /]# getenforce #查看当前运行模式

Enforcing

[root@svr7 /]# setenforce 0 #修改当前运行模式

[root@svr7 /]# getenforce

Permissive

[root@svr7 /]# vim /etc/selinux/config

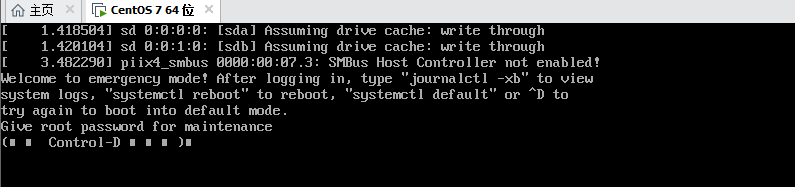
SELINUX=permissive

虚拟机B：同上

**六、系统故障修复**

* **如果/etc/fstab文件有误：修复办法**

1.输入root的密码



2.修改/etc/fstab文件内容

C:\Users\Python\AppData\Local\YNote\data\niubenlinux@163.com\09244c1a8bf449adaa13909084ae7e78\clipboard.png

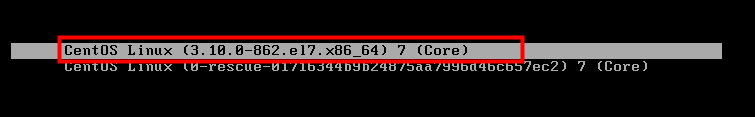
* **破解root密码思路**

前提：必须是服务器的管理者，涉及重启服务器

1)**重启系统**,进入 恢复模式（救援模式）

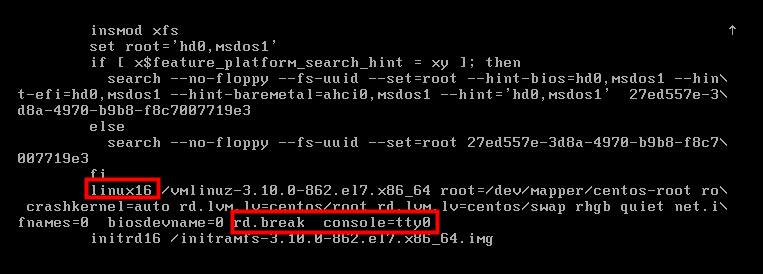
[root@A ~]# reboot

开启虚拟机A，在此界面按e键



找到linux16该行

在linux16该行的最后，空格输入 **rd.break console=tty0**



按 **ctrl + x** 启动，会看到switch\_root#

2)以可写方式重新挂载 /,并切换到此环境

# **chroot  /sysroot** #切换环境，切换到硬盘操作系统的环境

# **mount  -o  remount,rw   /**  #让根目录下所有数据，可以读也可以写入

3)重新设置root的密码

sh-4.2# **echo 1 | passwd --stdin  root**

4)如果SELinux是强制模式，才需要重设SELinux策略（其他模式不需要做此操作）

sh-4.2# vim /etc/selinux/config #查看SELinux开机的运行模式

sh-4.2# touch  /.autorelabel #让SELinux失忆

5)强制重启系统完成修复

sh-4.2# **reboot -f**

**七、构建基本服务**

* **构建Web服务**

Web服务：提供一个页面内容的服务

提供Web服务的软件：httpd、Nginx、tomcat

http协议：超文本传输协议

daemon

英 [ˈdiːmən] 美 [ˈdiːmən]

n.(古希腊神话中的)半神半人精灵

虚拟机A：

1.安装软件包

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

2.运行提供Web服务程序

]# > /etc/resolv.conf #清空此文件内容

]# **/usr/sbin/httpd** #绝对路径运行程序

]# pgrep -l httpd #查看进程信息

4.书写一个页面文件

[root@svr7 ~]# vim /var/www/html/index.html

哈哈嘻嘻呵呵~~~~~~~

5.浏览器访问测试

[root@svr7 ~]# curl http://192.168.4.7

哈哈嘻嘻呵呵~~~~~~~

* **构建FTP服务**

FTP：文本传输协议

实现FTP服务功能的软件：vsftpd

默认共享数据的主目录：/var/ftp/

虚拟机A

1.安装软件包

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

2.运行程序

[root@svr7 ~]# /usr/sbin/vsftpd

3.访问测试

[root@svr7 ~]# curl ftp://192.168.4.7

**八、防火墙的策略管理**

作用：隔离，严格过滤入站，放行出站

硬件防火墙

软件防火墙

* 系统服务：firewalld
* 管理工具：firewall-cmd、firewall-config
* 根据所在的网络场所区分，预设保护规则集
  + public：仅允许访问本机的ssh、dhcp、ping服务
  + trusted：允许任何访问
  + block：拒绝任何来访请求，明确拒绝客户端
  + drop：丢弃任何来访的数据包，不给任何回应
* 防火墙判定原则：

1.查看客户端请求中来源IP地址，查看自己所有区域中规则，那个区域中有该源IP地址规则，则进入该区域

2.进入默认区域（默认情况下为public）

* **防火墙默认区域的修改**

虚拟机A

]# firewall-cmd --get-default-zone #查看默认区域

虚拟机B

]# curl http://192.168.4.7 #失败

]# curl ftp://192.168.4.7 #失败

]# ping -c 2 192.168.4.7 #成功

虚拟机A：修改默认区域

]# firewall-cmd --set-default-zone=trusted

虚拟机B

]# curl http://192.168.4.7 #成功

]# curl ftp://192.168.4.7 #成功

* **防火墙public区域添加规则**

虚拟机A：添加允许的协议

]# firewall-cmd --set-default-zone=public

]# firewall-cmd --zone=public --add-service=http

]# firewall-cmd --zone=public --list-all

虚拟机B

]# curl http://192.168.4.7 #成功

]# curl ftp://192.168.4.7 #失败

虚拟机A：添加允许的协议

]# firewall-cmd --zone=public --add-service=ftp

]# firewall-cmd --zone=public --list-all

虚拟机B

]# curl http://192.168.4.7 #成功

]# curl ftp://192.168.4.7 #成功

* **防火墙public区域添加规则（永久）**

-永久（--permanent）

]# firewall-cmd --reload #加载防火墙永久策略

]# firewall-cmd --zone=public --list-all

]# firewall-cmd --permanent --zone=public --add-service=http

]# firewall-cmd --permanent --zone=public --add-service=ftp

]# firewall-cmd --zone=public --list-all

]# firewall-cmd --reload #加载防火墙永久策略

]# firewall-cmd --zone=public --list-all

课后习题：

案例1：设置SELinux保护

为虚拟机A、虚拟机B 配置SELinux

1）确保 SELinux 处于宽松模式（permissive）

2）在每次重新开机后，此设置必须仍然有效

案例2：实现虚拟机A 的Web服务

1）利用httpd软件搭建Web服务，页面显示内容为 小蝌蚪找妈妈

案例3：实现虚拟机A 的防火墙配置

1）修改虚拟机A防火墙配置，明确拒绝所有客户端访问（将默认区域设置为block）

2）在虚拟机B上,测试能否访问虚拟机A 的Web服务

3）在虚拟机B上,测试能否 ping通 虚拟机A

案例4：实现虚拟机A 的防火墙配置

1）修改虚拟机A防火墙配置，将默认区域修改为public

2）在虚拟机B上,测试能否访问虚拟机A的Web服务

3）在虚拟机B上,测试能否 ping通 虚拟机A

案例5：实现虚拟机A的防火墙配置

1）修改虚拟机A防火墙配置，将默认区域修改为public

2）修改虚拟机A永久的防火墙配置，在public区域中添加http协议

3）在虚拟机B上,测试能否访问虚拟机A 的Web服务

前面习题：

案例1:计划任务

-指定一个计划任务，以root身份，每2分钟记录当前的系统时间，写入到/opt/two.txt

[root@A ~]# crontab -e

\*/2 \* \* \* \* date >> /opt/two.txt

[root@A ~]# crontab -l

[root@A ~]# cat /opt/two.txt

案例2：chmod权限设置

1）以root用户新建/nsddir/目录，在该目录下新建文件readme.txt

2）使用户zhangsan能够在/nsddir/目录下创建/删除子目录

3）使用户zhangsan能够修改/nsddir/readme.txt文件的容

[root@A ~]# mdkir /nsddir

[root@A ~]# echo haha >> /nsddir/readme.txt

[root@A ~]# chmod o+w /nsddir

[root@A ~]# su - zhangsan

[zhangsan@A ~]$ mkdir /nsddir/zhangsan

[zhangsan@A ~]$ ls /nsddir

[zhangsan@A ~]$ exit

[root@A ~]# chmod o+w /nsddir/readme.txt

[root@A ~]# su - zhangsan

[zhangsan@A ~]$ echo xixi >> /nsddir/readme.txt

[zhangsan@A ~]$ cat /nsddir/readme.txt

[zhangsan@A ~]$ exit

案例3：chown归属设置

1）新建/tarena1目录

a）将属主设为gelin01，属组设为tarena组

b）使用户gelin01对此目录具有rwx权限，其他人对此目录无任何权限

[root@A ~]# mkdir /tarena1

[root@A ~]# useradd gelin01

[root@A ~]# groupadd tarena

[root@A ~]# chown gelin01:tarena /tarena1

[root@A ~]# ls -ld /tarena1

[root@A ~]# chmod o=--- /tarena1

[root@A ~]# ls -ld /tarena1

2）使用户gelin02能进入、查看/tarena1文件夹（提示：将gelin02加入所属组）

[root@A ~]# useradd gelin02

[root@A ~]# gpasswd -a gelin02 tarena

[root@A ~]# id gelin02

[root@A ~]# su - gelin02

[gelin02@A ~]$ cd /tarena1

[gelin02@A tarena1]$ cd /tarena1

[gelin02@A tarena1]$ ls

[gelin02@A tarena1]$ exit

3）新建/tarena2目录

a）将属组设为tarena

b）使tarena组的任何用户都能在此目录下创建、删除文件

[root@A ~]# mkdir /tarena2

[root@A ~]# chown :tarena /tarena2

[root@A ~]# chmod g+w /tarena2

[root@A ~]# ls -ld /tarena2

[root@A ~]# useradd ceshi

[root@A ~]# gpasswd -a ceshi tarena

[root@A ~]# id ceshi

[root@A ~]# su - ceshi

[ceshi@A ~]$ mkdir /tarena2/ceshi

[ceshi@A ~]$ ls /tarena2

[ceshi@A ~]$ exit

4）新建/tarena/public目录

a）使任何用户对此目录都有rwx权限

b）拒绝zhangsan进入此目录，对此目录无任何权限

[root@A ~]# mkdir /tarena/public

[root@A ~]# chmod 777 /tarena/public

[root@A ~]# ls -ld /tarena/public

[root@A ~]# setfacl -m u:zhangsan:--- /tarena/public

[root@A ~]# su - zhangsan

[zhangsan@A ~]$ ls /tarena/public

[zhangsan@A ~]$ cd /tarena/public

[zhangsan@A ~]$ exit

案例4:权限设置

1、创建文件夹/data/test,设置目录的访问权限，使所有者和所属组具备读写执行的权限；其他人无任何权限。

2、递归修改文件夹/data/test的归属使所有者为zhangsan，所属组为tarena。

3、请实现在test目录下，新建的所有子文件或目录的所属组都会是tarena。

4、为lisi创建ACL访问权限，使得lisi可以查看/etc/shadow文件

[root@A ~]# mkdir /data/test

[root@A ~]# chmod u=rwx,g=rwx,o=--- /data/test 或者 chmod 770 /data/test

[root@A ~]# ls -ld /data/test

[root@A ~]# chown -R zhangsan:tarena /data/test

[root@A ~]# ls -ld /data/test

[root@A ~]# chmod g+s /data/test

[root@A ~]# mkdir /data/test/abc

[root@A ~]# ls -ld /data/test/abc

[root@A ~]# setfacl -m u:lisi:r /etc/shadow

[root@A ~]# getfacl /etc/shadow

[root@A ~]# su - lisi

[lisi@A ~]$ cat /etc/shadow

[lisi@A ~]$ exit

案例5:虚拟机 server0上操作

将文件 /etc/fstab 拷贝为 /var/tmp/fstab，并调整文件 /var/tmp/fstab权限

满足以下要求：

– 此文件的拥有者是 root

– 此文件对任何人都不可执行

– 用户 natasha 能够对此文件执行读和写操作

– 用户 harry 对此文件既不能读，也不能写

[root@A ~]# cp /etc/fstab /var/tmp/fstab

[root@A ~]# ls -l /var/tmp/fstab

[root@A ~]# setfacl -m u:natasha:rw /var/tmp/fstab

[root@A ~]# getfacl /var/tmp/fstab

[root@A ~]# su - natasha

[natasha@A ~]$ cat /var/tmp/fstab

[natasha@A ~]$ echo ceshi >> /var/tmp/fstab

[natasha@A ~]$ cat /var/tmp/fstab

[natasha@A ~]$ exit

[root@A ~]# setfacl -m u:harry:--- /var/tmp/fstab

[root@A ~]# getfacl /var/tmp/fstab

[root@A ~]# su - harry

[harry@A ~]$ cat /var/tmp/fstab

[harry@A ~]$ echo ceshi >> /var/tmp/fstab

[harry@A ~]$ exit

案例6:虚拟机操作

创建一个共用目录 /home/admins，要求如下：

– 此目录的所属组是 adminuser

– adminuser 组的成员对此目录有读写和执行的权限，并且其他用户没有任何权限

– 在此目录中创建的文件，其所属组会自动设置为 属于 adminuser 组

[root@A ~]# mkdir /home/admins

[root@A ~]# groupadd adminuser

[root@A ~]# chown :adminuser /home/admins

[root@A ~]# ls -ld /home/admins

[root@A ~]# chown g+s /home/admins

[root@A ~]# ls -ld /home/admins

[root@A ~]# mkdir /home/admins/ceshi

[root@A ~]# ls -ld /home/admins/ceshi

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

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

[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

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

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 server0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？

5.开启5个xeyes放入后台运行

6.杀死所有xeyes进程

[root@A ~]# pstree -ap

[root@A ~]# pstree -ap lisi

[root@A ~]# ps aux

[root@A ~]# pgrep cron

[root@A ~]# xeyes &

[root@A ~]# xeyes &

[root@A ~]# xeyes &

[root@A ~]# xeyes &

[root@A ~]# xeyes &

[root@A ~]# killall xeyes

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

案例2: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

[root@B ~]# hostname test.example.com

[root@B ~]# echo test.example.com > /etc/hostname

[root@B ~]# cat /etc/hostname

[root@B ~]# nmcli connection show

[root@B ~]# nmcli connection modify 'ens33' ipv4.method manual ipv4.addresses 172.25.0.11/18 ipv4.gateway 172.25.0.254 connection.autoconnect yes

[root@B ~]# nmcli connection up 'ens33'

[root@B ~]# route -n #查看网关地址

[root@B ~]# ifconfig | head -2 #查看网卡地址

[root@B ~]# echo nameserver 172.25.254.254 > /etc/resolv.conf

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

案例3: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

[root@B ~]# hostname B.tedu.cn

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

[root@B ~]# cat /etc/hostname

[root@B ~]# nmcli connection show

[root@B ~]# nmcli connection modify 'ens33' ipv4.method manual ipv4.addresses 192.168.1.1/24 ipv4.gateway 192.168.1.254 connection.autoconnect yes

[root@B ~]# nmcli connection up 'ens33'

[root@B ~]# route -n #查看网关地址

[root@B ~]# ifconfig | head -2 #查看网卡地址

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

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

案例4: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

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

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

[root@B ~]# cat /etc/hostname

[root@B ~]# nmcli connection show

[root@B ~]# nmcli connection modify 'ens33' ipv4.method manual ipv4.addresses 192.168.4.207/24 ipv4.gateway 192.168.4.254 connection.autoconnect yes

[root@B ~]# nmcli connection up 'ens33'

[root@B ~]# route -n #查看网关地址

[root@B ~]# ifconfig | head -2 #查看网卡地址

[root@B ~]# echo nameserver 1.1.1.1 > /etc/resolv.conf

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

案例5: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

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

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

[root@A ~]# cat /etc/hostname

[root@A ~]# nmcli connection show

[root@A ~]# nmcli connection modify 'ens33' ipv4.method manual ipv4.addresses 192.168.4.7/24 ipv4.gateway 192.168.4.254 connection.autoconnect yes

[root@A ~]# nmcli connection up 'ens33'

[root@A ~]# route -n #查看网关地址

[root@A ~]# ifconfig | head -2 #查看网卡地址

[root@A ~]# echo nameserver 1.1.1.1 > /etc/resolv.conf

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

案例6：虚拟机B

1.源码编译安装 inotify-tools 软件工具

2.安装位置为/usr/local/tools

[root@svr7 ~]# yum -y install gcc make

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

[root@svr7 ~]# tar -xf /tools/inotify-tools-3.13.tar.gz -C /

[root@svr7 ~]# ls /

[root@svr7 ~]# ls /inotify-tools-3.13/

[root@svr7 ~]# cd /inotify-tools-3.13/

[root@svr7 inotify-tools-3.13]# ./configure --prefix=/usr/local/tools #指定安装位置

[root@svr7 inotify-tools-3.13]# make

[root@svr7 inotify-tools-3.13]# make install

[root@svr7 inotify-tools-3.13]# ls /usr/local/tools

案例7：虚拟机B：自定义Yum仓库

1.将tools.tar.gz释放到/usr/local目录下

2.利用/usr/local/tools/other目录下RPM软件包，构建自定义Yum仓库

[root@pc207 ~]# tar -xf /root/tools.tar.gz -C /usr/local

[root@pc207 ~]# ls /usr/local/tools/other

[root@pc207 ~]# createrepo /usr/local/tools/other

[root@pc207 ~]# ls /usr/local/tools/other

[root@pc207 ~]# vim /etc/yum.repos.d/dvd.repo

[dvd]

name=CentOS7

baseurl=file:///dvd

enabled=1

gpgcheck=0

[test]

name=myrpm

baseurl=file:///usr/local/tools/other

enabled=1

gpgcheck=0

[root@pc207 ~]# yum repolist

案例8：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

[root@pc207 ~]# scp -r /usr/local/tools/other root@192.168.4.7:/usr

[root@pc207 ~]# scp /etc/gshadow root@192.168.4.7:/root

[root@pc207 ~]# scp -r /etc/skel root@192.168.4.7:/tmp

虚拟机A

[root@svr7 ~]# ls /usr

[root@svr7 ~]# ls /root

[root@svr7 ~]# ls /tmp

案例9：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

1.生成公钥 私钥

[root@pc207 ~]# ssh-keygen #一路回车

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

2.传递公钥 到虚拟机A

[root@pc207 ~]# ssh-copy-id root@192.168.4.7

3.虚拟机A：查看

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

4.虚拟机B：测试无密码

[root@pc207 ~]# ssh root@192.168.4.7