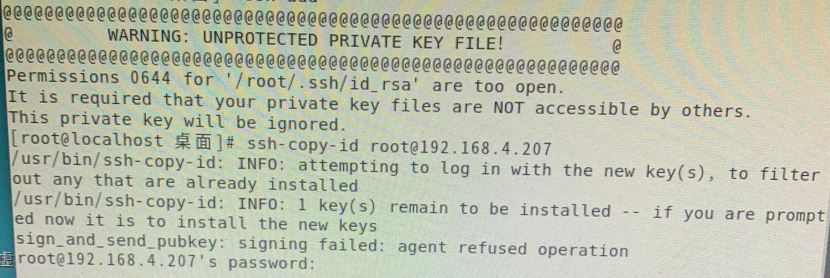
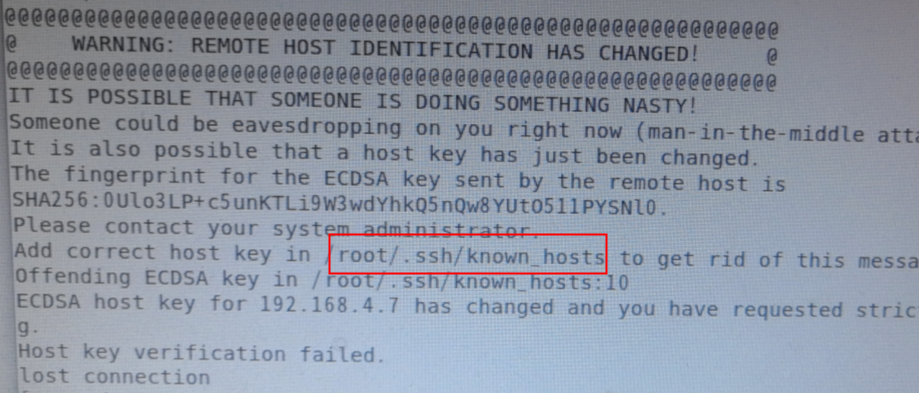
常见的SSH报错：

1.ssh的秘钥文件权限不是600



2.ssh远程管理与记录不同(删除/root/.ssh/known\_hosts)



**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

3.删除添加IP地址

]# ip address del 192.168.10.1/24 dev eth0

* **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

**一、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：实时跟踪日志消息**

[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

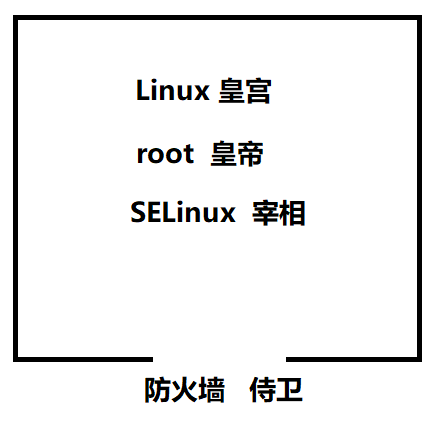
6 INFO（信息） 一般信息

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**

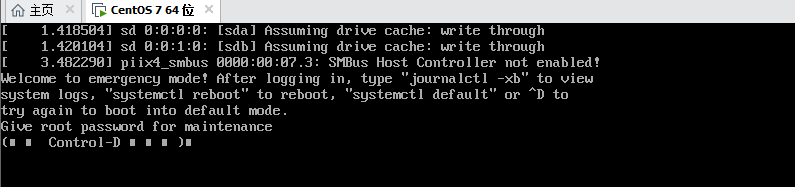
SELINUX=**permissive**

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

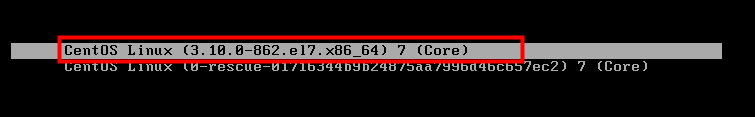
* **破解root密码思路**

前提：必须是服务器的管理者，涉及重启服务器

1)**重启系统**,进入救援模式

]# reboot

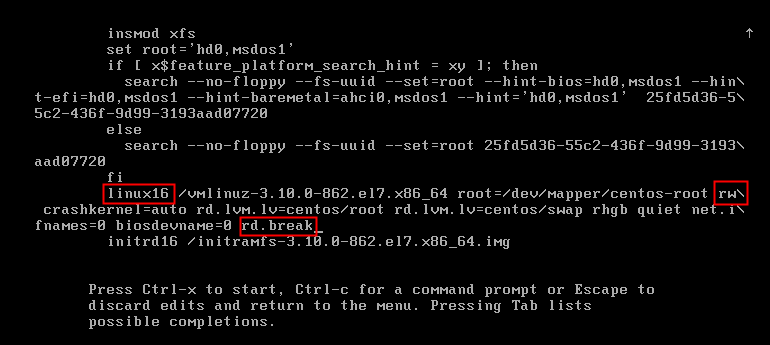
开启虚拟机A，在此界面按e键



找到linux16该行

在linux16该行找到**ro**将其修改为**rw**

在linux16该行最后空格输入 **rd.break**



按 **ctrl + x** 启动，会看到**switch\_root#**

2)切换到此环境

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

**switch\_root# ls #查看硬盘系统的数据**

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（SELinux是强制模式）

5)强制重启系统完成修复

sh-4.2# **reboot -f**

**七、构建基本服务**

**虚拟机A：svr7机器**

* **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 （守护神）

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~~~~~~~

* **虚拟机A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/

1.安装软件包

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

2.运行程序

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

[root@svr7 ~]# **pgrep -l 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---->iptables
* 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**

* **防火墙public区域删除规则（永久）**

]# **firewall-cmd --reload**  #加载防火墙永久策略

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

]# **firewall-cmd --permanent --zone=public --remove-service=http**

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

]# **firewall-cmd --reload** #加载防火墙永久策略

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

**九、服务的管理**

用户---》systemd---》服务

•上帝进程：systemd

•Linux系统和服务管理器

–是内核引导之后加载的**第一个初始化进程**（PID=1）

–负责掌控整个Linux的运行/服务资源组合

•一个更**高效**的系统&服务管理器

–开机服务**并行启动**，各系统服务间的精确依赖

–配置目录：/etc/systemd/system/

–服务目录：/lib/systemd/system/

–主要管理工具：systemctl

[root@svr7 ~]# systemctl -t service --all #列出所有的服务

•对于服务的管理（与手动启动有冲突）

**systemctl restart 服务名** #重起服务

systemctl start 服务名 #开启服务

systemctl stop 服务名 #停止服务

systemctl status 服务名 #查看服务当前的状态

**systemctl enable 服务名**  #设置服务开机自启动

systemctl disable 服务名 #设置服务禁止开机自启动

systemctl is-enabled 服务名 #查看服务是否开机自启

]# **yum -y install httpd**

]# **> /etc/resolv.conf**  #清除文件内容

]# **killall httpd**  #杀死手动启动的httpd

]# **systemctl restart httpd**  #重启httpd服务

]# **systemctl status httpd**  #查看服务httpd状态

]# **systemctl enable httpd**  #设置httpd开机自启动

Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.

]# **systemctl is-enabled httpd** #查看httpd是否是开机自启动

]# **systemctl disable httpd** #关闭httpd开机自启动

Removed symlink /etc/systemd/system/multi-user.target.wants/httpd.service.

]# **systemctl is-enabled httpd** #查看httpd是否是开机自启动

**管理运行级别**

RHEL6:运行级别 300

0：关机 0个服务

1：单用户模式（基本功能的实现，破解Linux密码） 50个服务

2：多用户字符界面（不支持网络） 80个服务

**3：多用户字符界面（支持网络）服务器默认运行级别** 100个服务

4：未定义 0个服务

**5：图形界面 300个服务**

6：重起 0个服务

切换运行级别：**init** 数字

**RHEL7：运行模式（运行级别）**

字符模式：multi-user.target

图形模式：graphical.target

当前直接切换到字符模式

]# **systemctl isolate multi-user.target** #相当于原来的init 3

当前直接切换到图形模式

]# **systemctl isolate graphical.target** #相当于原来的init 5

查看每次开机默认进入模式

[root@svr7 /]# **systemctl get-default**

设置永久策略，每次开机自动进入multi-user.target

[root@svr7 /]# **systemctl set-default multi-user.target**

[root@svr7 /]# **reboot**

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：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

前面习题：

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