**一、Admin**

**2018-06-29(第一天)**

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

TCP/IP协议及配置

vnc服务器地址：172.40.50.114：7000

TCP/IP协议简介

• TCP/IP是最广泛支持的通信协议集合

– 包括大量Internet应用中的标准协议

– 支持跨网络架构、跨操作系统平台的通信

• **主机与主机之间通信的三个要素**

**– IP地址(IP address)**

**– 子网掩码(subnet mask)**

**– IP路由(IP router) 路由器**

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

– IP地址(IP address)

作用：唯一标识网络中一台主机

**一共32个二进制位，用点划分了4部分，最终用4个十进制数表示**

**11111111.00000000.11111111.00000000**

**Windows2008密码：Taren1**

**• 用于一般计算机网络**

**– A类:1 ~ 127 网+主+主+主**

**– B类:128 ~ 191 网+网+主+主**

**– C类:192 ~ 223 网+网+网+主**

**• 组播及科研专用**

**– D类:224 ~ 239 组播**

**– E类:240 ~ 254 科研**

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

IP地址的组成：网络位+主机位

192.168.1.1------>IP地址类别（C）------>网+网+网+主

192.168.1.1：在192.168.1网络里，编号为1的主机

192.168.1.2：在192.168.1网络里，编号为2的主机

172.10.2.1：在172.10网络里，编号为2.1的主机

网络位：标识一个网络（区号）

主机位：网络中的编号

010 12345678 ----- 北京

区号 号码

0316 7351234 ----- 廊坊

– 子网掩码(subnet mask):用来标识IP地址网络位与主机位

用1标识网络位，用0主机位

192.168.1.1

11000000.10101000.00000001.00000001

子网掩码 11111111.11111111.11111111.00000000

24个网络位.8个主机位

192.168.1.1 255.255.255.0

192.168.1.1/24 24个网络位

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

配置IP地址

– 右击桌面“网络” ---> 属性 --->更改适配器设置

– 双击“本地连接” ---> 属性

– 双击“Internet协议版本4 (TCP/IPv4)” --------> “使用下面的IP地址”

– 配置完成后,单击“确定”完成

查看IP地址：

1.图形查看：

右击桌面“网络” ---> 属性 --->更改适配器设置（双击）--->详细信息

2.命令查看：ipconfig

windows键 + r键

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

虚拟化软件：虚拟计算机硬件，用真的虚拟出假的，最终假的能够当真的来用

1.Windows：虚拟化软件VMware

2.Linux：虚拟化软件KVM

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

关闭Windows2008进行克隆

双击 桌面 虚拟系统管理器---->右击 “Windows2008”---->克隆

克隆之后的虚拟机配置IP地址：192.168.1.2/24

配置IP地址

– 右击桌面“网络” ---> 属性 --->更改适配器设置

– 双击“本地连接” ---> 属性

– 双击“Internet协议版本4 (TCP/IPv4)” --------> “使用下面的IP地址”

– 配置完成后,单击“确定”完成

1.命令查看：ipconfig

windows键 + r键

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

Windows2008 修改桌面背景：开始----->搜索----->desktop

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

测试通信命令ping,一次ping操作是双向的，有来有回

两台虚拟机均，关闭Windows2008防火墙：

– 右击桌面“网络” ---> 属性 --->左下角 Windows防火墙---->打开或关闭防火墙

windows键 + r键 --------> ping 192.168.1.2

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

**2018-06-30（第二天）**

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

IP地址配置方式：

1.手动配置

2.DHCP自动获取IP地址（前提网络里必须要有DHCP服务器）

DNS服务器：将域名解析为对应IP地址

为了方便记忆网站，网站的推广

网关地址：一个网络到另一个网络的关口（路由器）

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

**Linux是一种操作系统**

**操作系统：一堆软件的集合，计算机硬件正常合理工作**

**Unix/Linux发展史**

**• UNIX诞生, 1970-1-1**

**Linux的诞生，Linus Torwalds**

Linux系统内核

版本号:主版本.次版本.修订号

内核：调配所有的硬件

**用户 --------> 内核 --------> 硬件**

• 是一套公开发布的完整Linux系统

– Linux内核 + 各种应用程序

**Linux发行版本**

**• 发行版的名称/版本由发行方决定**

**– Red Hat Enterprise Linux 5/6/7**

**– Suse Linux Enterprise 12**

**– Debian Linux 7.8**

**– Ubuntu Linux 14.10/15.04**

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

虚拟机安装RHEL7.4操作系统

1.虚拟计算的硬件

Ctrl+Alt ：鼠标回到真机

软件包选择：带GUI的服务器

如何使用硬盘?

• 一块硬盘的“艺术”之旅

– 物理硬盘==>分区规划==>格式化==>读/写文档

– 毛坯楼层 ==>打隔断==>装修==>入驻

格式化： 数据在空间，存储的规则 （文件系统）

**文件系统： ext4(RHEL6) xfs(RHEL7)**

**SWAP,交换空间(虚拟内存)：缓解真实物理内存的压力**

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

Linux目录结构：树型结构

• 最顶层为根目录(/)

**根目录(/)：Linux所有数据都在此目录下（Linux系统的起点）**

路径：/root/abc/test.txt

/dev：所有设备相关的文件

硬盘标识的方法：

**hd,表示IDE设备**

**sd,表示SCSI设备**

**vd,表示虚拟化设备（只在虚拟机里出现**）

/dev/hda1 /dev/hdb2 /dev/hdc3 /dev/hdd1

/dev/sda1

**/dev/sdb2 : SCSI类型的设备，第二块硬盘，第二个分区**

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

命令行基本操作

• 控制台切换 ( Ctrl + Alt + Fn 组合键)

– tty1:图形桌面

– tty2 ~ tty6:字符控制台

图形命令行：右键“ 打开终端”

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

• 命令行提示标识的含义

– [当前用户@主机名 工作目录]$

– 若当前用户是root,则最后一个字符为 #

[root@svr7 桌面]#

– 否则,最后一个字符为 $

[teacher@svr7 桌面]$

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

查看及切换目录

• pwd — Print Working Directory

– 用途:显示当前所在位置

• cd — Change Directory

– 用途:切换工作目录

– 格式:cd [目标文件夹位置]

• ls — List

– 格式:ls [选项]... [目录或文件名]...

**蓝色:目录**

**黑色：文本文件**

**青色：快捷方式**

**绿色：可执行文件**

[root@localhost /]# pwd #显示当前所在路径

[root@localhost /]# cd / #切换到根目录下

[root@localhost /]# pwd

[root@localhost /]# ls #查看当前路径内容

[root@localhost /]# cd /boot

[root@localhost boot]# pwd

[root@localhost boot]# ls

[root@localhost boot]# cd /dev

[root@localhost dev]# pwd

[root@localhost dev]# ls

[root@localhost dev]# cd /etc

[root@localhost etc]# pwd

[root@localhost etc]# ls

[root@localhost etc]# ls /root

快捷键：Ctrl + l 清屏

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

cat：查看文本文件内容

**[root@localhost /]# cat /etc/redhat-release #查看系统版本**

Red Hat Enterprise Linux Server release 7.4

[root@localhost /]# cat /root/anaconda-ks.cfg

[root@localhost /]# cat /etc/passwd

**• 命令行的一般格式**

**– 命令字 [选项]... [参数1] [参数2]**...

cat -n /etc/redhat-release

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

• ls — List

– 格式:ls [选项]... [目录或文件名]...

• 常用命令选项

– -l :以长格式显示 ,显示目录内容的详细信息

[root@localhost /]# ls -l /root

[root@localhost /]# cat -n /etc/redhat-release

**[root@localhost /]# uname -r #列出内核版本**

[root@localhost /]# ls -l /boot

[root@localhost /]# ls -l /etc/redhat-release

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

请在真机上操作

**• 列出CPU处理器信息**

**[root@svr7 桌面]# lscpu**

**• 检查内存大小、空闲情况**

**[root@svr7 桌面]# cat /proc/meminfo**

• 列出当前系统的主机名称

[root@A ~]# hostname

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

[root@A ~]# hostname

打开新的图形终端窗口，会刷新主机名

• 列出IP信息

[root@A ~]# ifconfig

[root@A ~]# ifconfig eth0 192.168.1.1 #为eth0临时配置IP地址

[root@A ~]# ifconfig

[root@A ~]# ping 192.168.1.1

**127.0.0.1：永远代表本机**

[root@A ~]# ping 127.0.0.1

Ctrl + c：结束正在运行的指令

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

• 关机:poweroff

[root@svr7 ~]# poweroff

• 重启:reboot

[root@svr7 ~]# reboot

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

• mkdir — Make Directory

– 格式:mkdir [/路径/]目录名...

[root@A ~]# mkdir /opt/nsd01

[root@A ~]# ls /opt/

[root@A ~]# mkdir /root/nsd02 /mnt/nsd03

[root@A ~]# ls /root/

[root@A ~]# ls /mnt/

• touch命令

– 用途:新建空文件

– 格式:touch 文件名...

[root@A opt]# touch /opt/1.txt

[root@A opt]# ls /opt/

[root@A opt]# touch /opt/2.txt /mnt/3.txt

[root@A opt]# ls /opt/

[root@A opt]# ls /mnt/

专门用于查看内容较多的文件

[root@A opt]# less /etc/passwd

按上 下键可以滚动

输入 /root 可以全文查找root

按 q 退出

• head、tail 命令

– 格式:

head -数字 文件名

tail -数字 文件名

[root@A opt]# head -1 /etc/passwd

[root@A opt]# head -2 /etc/passwd

[root@A opt]# tail -1 /etc/passwd

[root@A opt]# tail -2 /etc/passwd

[root@A opt]# head /etc/passwd #默认显示前10行

[root@A opt]# tail /etc/passwd #默认显示后10行

[root@A opt]# head -12 /etc/passwd #显示前12行

[root@A opt]# tail -13 /etc/passwd #显示后13行

• grep工具

– 用途:输出包含指定字符串的行

– 格式:grep 字符串 目标文件

[root@A opt]# grep root /etc/passwd

[root@A opt]# grep bin /etc/passwd

[root@A opt]# grep sbin /etc/passwd

[root@A opt]# grep lisi /etc/passwd

[root@A opt]# grep -n root /etc/passwd #显示原文行号

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

以 / 开始的绝对路径

以当前为参照的相对路径

[root@A /]# cd /etc/pki/

[root@A pki]# ls

[root@A pki]# cd /etc/pki/java

[root@A java]# pwd

[root@A java]# cd /etc/pki/

[root@A pki]# pwd

[root@A pki]# ls

[root@A pki]# cd java

[root@A java]# pwd

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

**2018-07-02（第三天）**

**早晨默写作业：**

1. Linux目录结构中“/”与“/dev”主要存放的内容？

/:所有数据都在此目录下(Linux系统的起点)

/dev:设备相关数据

2. Linux中磁盘的表示，第三块SCSI磁盘的第二个分区？

/dev/sdc2

3. Linux中常见的文件系统都有哪些？

ext4 xfs swap

4. Linux中查看文本文件命令是？

cat less

5. Linux中查看目录内容的命令是？

ls

6. Linux中查看主机名的命令是？

hostname

7. Linux中查看IP地址的命令是？

ifconfig

8. Linux中切换到/dev目录的命令是？

cd /dev

9. Linux中显示当前位置的命令是？

pwd

10.Linux中显示/root目录内容详细属性怎么实现？

ls -l /root

11.Linux中显示当前主机的内核版本的命令？

uname -r

12.Linux中查看CPU信息命令是？

lscpu

13.Linux中查看内存信息命令是？

cat /proc/meminfo

14.Linux中查看/boot目录内容如何操作？

ls /boot

15.Linux中查看/etc/passwd文件前两行如何操作？

head -2 /etc/passwd

16.Linux中查看/etc/passwd/中包含root的行，如何操作？

grep root /etc/passwd

• Linux命令

– 用来实现某一类功能的指令或程序

(执行命令时,Linux需要找到对应的执行文件)

– 命令的执行依赖于解释器(默认解释器:/bin/bash)

用户--------> 解释器 --------> 内核 ------->硬件

绿色:可以执行的程序

命令行的一般格式

• 基本用法

– 命令字 [选项]... [参数1] [参数2]...

快速编辑技巧

• Tab键自动补全

– 可补齐命令字、选项、参数、文件路径、软件名、服务名

# ls /etc/sysconfig/network-scripts/ifcfg-eth0

# ls /et(tab)/sysco(tab)/netw(tab)-(tab)/ifc(tab)-e(tab)

# cat /et(tab)/red(tab)-r(tab)

# if(tab)(tab) #连续按两次tab

# ifco(tab)

# ls /e(tab)/pas(tab)

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

• 快捷键

– Ctrl + l :清空整个屏幕

– Ctrl + c :结束正在运行的程序

– Esc + . 或 Alt + . :粘贴上一个命令的参数

# ls -l /etc/sysconfig/network-scripts/ifcfg-eth0

# cat Alt + .

# cat -n Alt + .

– Ctrl + u:清空至行首

– Ctrl + w:往回删除一个单词(以空格界定)

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

mount挂载操作:让目录成为设备的访问点

一 显示光盘所有内容

Windows:

光盘------->光驱设备------->CD驱动器(图标)

Linux:

光盘------->光驱设备---------->目录(手动) 访问点\挂载点

/dev/cdrom

访问点:访问设备资源时候,通过的点

煤矿----------->洞口

1.图形操作,利用鼠标,将光盘文件放入虚拟光驱设备

2.查看光驱设备

[root@A ~]# ls /dev/cdrom

/dev/cdrom

[root@A ~]# ls -l /dev/cdrom

lrwxrwxrwx. 1 root root 3 7月 2 10:36 /dev/cdrom -> sr0

[root@A ~]# ls /dev/sr0

3.创建目录

[root@A ~]# mkdir /dvd

[root@A ~]# ls /dvd

• 使用mount命令

– 格式:mount 设备路径 挂载点目录

[root@A ~]# mount /dev/cdrom /dvd/

mount: /dev/sr0 写保护，将以只读方式挂载

[root@A ~]# ls /dvd/

[root@A ~]# ls /dvd/Packages

4.卸载

[root@A ~]# umount /dvd

[root@A ~]# ls /dvd/

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

常见的报错:

1.卸载:

[root@A dvd]# umount /dvd

umount: /dvd：目标忙。

2.挂载建议挂载到自己创建的目录

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

• cd — Change Directory

– 用途:切换工作目录

– 格式:cd [目标文件夹位置]

~用户名 表示用户 用户名 的家目录

/root: 管理员的家目录

/home: 存放所有普通用户家目录的地方

[root@A /]# cd ~root

[root@A ~]# pwd

[root@A ~]# useradd zhangsan

[root@A ~]# cd ~zhangsan

[root@A zhangsan ]# pwd

[root@A zhangsan ]# useradd tom

[root@A zhangsan ]# cd ~tom

[root@A tom ]# pwd

[root@A tom ]# useradd dc

[root@A tom ]# cd ~dc

[root@A dc ]# pwd

. 表示当前目录

.. 表示父目录

[root@A /]# cd /etc/pki/

[root@A pki]# pwd

[root@A pki]# cd ..

[root@A etc]# pwd

[root@A etc]# cd ..

[root@A /]# pwd

[root@A /]# cd /etc/pki/

[root@A pki]# pwd

[root@A pki]# cd ../..

[root@A /]# pwd

[root@A /]# cd /var/log

[root@A log]# pwd

[root@A log]# cd ..

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

ls 列出文档及属性

• ls — List

– 格式:ls [选项]... [目录或文件名]

**• 常用命令选项**

**– -l:以长格式显示**

**– -A:包括名称以 . 开头的隐藏文档**

**– -d:显示目录本身(而不是内容)的属性**

**– -h:提供易读的容量单位(K、M等)**

[root@A /]# ls --help

[root@A /]# ls /root/

[root@A /]# ls -A /root/

[root@A /]# ls -l /boot

[root@A /]# ls -lh /boot

[root@A /]# ls -lhd /boot

[root@A /]# ls -lhA /root/

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

**补充:统计目录的大小**

**[root@A /]# du -sh /root/ /etc/ /home/**

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

**使用通配符**

**– \*:任意多个任意字符**

**– ?:单个字符**

[root@A /]# ls /dev/tty\* #匹配/dev/目录下tty开头的

[root@A /]# ls /boot/vm\* #匹配/boot目录下vm开头的

[root@A /]# ls /etc/\*tab #匹配/etc目录下tab结尾的

[root@A /]# ls /etc/\*.conf #匹配/etc目录下.conf结尾的

[root@A /]# ls /dev/tty?

[root@A /]# ls /dev/tty??

**– [a-z]:多个字符或连续范围中的一个,若无则忽略**

**– {a,min,xy}:多组不同的字符串,全匹配**

[root@A /]# ls /dev/tty[3-9]

[root@A /]# ls /dev/tty[1-7]

[root@A /]# ls /dev/tty{1,3,5,10,27}

[root@A /]# ls /dev/tty{1,3,5,7,9}

[root@A /]# ls /dev/tty{S0,S1,S2}

请利用通配符,显示 /dev/tty20 至 /dev/tty30 ?

[root@A /]# ls /dev/tty{2[0-9],30}

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

别名的定义:简化复杂的命令

• 查看已设置的别名

– alias [别名名称]

• 定义新的别名

– alias 别名名称= '实际执行的命令行'

• 取消已设置的别名

– unalias [别名名称]

[root@A /]# alias hn='hostname' #定义别名

[root@A /]# hn

[root@A /]# alias #显示当前所有生效的别名

[root@A /]# unalias hn #删除别名

[root@A /]# hn

bash: hn: 未找到命令...

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

mkdir 创建目录

• mkdir — Make Directory

– 格式:mkdir [-p] [/路径/]目录名... #-p表示连父目录一起创建

[root@A /]# mkdir -p /vod/movie/cartoon

[root@A /]# mkdir -p /opt/aa/bb/cc/dd

[root@A /]# mkdir /root/stu01 /opt/stu02

[root@A /]# ls /root

[root@A /]# ls /opt

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

rm 删除

• rm — Remove

– 格式:rm [选项]... 文件或目录...

• 常用命令选项

– -r、-f:递归删除(含目录)、强制删除

递归:目录本身及目录下所有

[root@A /]# rm -rf /opt/\*

[root@A /]# touch /opt/1.txt

[root@A /]# touch /opt/2.txt

[root@A /]# mkdir -p /opt/tedu/abc

[root@A /]# ls /opt

[root@A /]# rm -rf /opt/1.txt

[root@A /]# ls /opt

[root@A /]# rm -rf /opt/2.txt

[root@A /]# ls /opt

[root@A /]# rm -rf /opt/tedu

[root@A /]# ls /opt

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

使用vim创建/修改文件

• vim文本编辑器

– 格式:vim [[/目录/]文件名]

**命令模式 输入模式(编辑模式) 末行模式**

[root@A /]# vim /opt/haxi.txt

命----- i键--------------->输入模式(按Esc 键回到命令模式)

令

模

式------ : 键-------------->末行模式(按Esc 键回到命令模式)

末行模式 :wq 保存并退出

末行模式 :q! 请值不保存退出

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

mv 移动/改名

• mv — Move

– 格式:mv 原文件... 目标路径

[root@A /]# rm -rf /opt/\*

[root@A /]# touch /opt/1.txt

[root@A /]# mkdir /opt/nsd

[root@A /]# ls /opt/

[root@A /]# mv /opt/1.txt /opt/nsd

[root@A /]# ls /opt/

[root@A /]# ls /opt/nsd/

重命名:路径不变的移动

[root@A /]# mv /opt/nsd/ /opt/student

[root@A /]# ls /opt/

[root@A /]# mv /opt/student/ /opt/abc

[root@A /]# ls /opt/

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

cp 复制

• cp — Copy

– 格式:cp [选项]... 原文件... 目标路径

• 常用命令选项

– -r:递归,复制目录时必须有此选项

[root@A /]# rm -rf /opt/\*

[root@A /]# cp /etc/redhat-release /opt/

[root@A /]# ls /opt/

[root@A /]# cp /etc/fstab /opt/

[root@A /]# ls /opt/

[root@A /]# cp -r /boot/ /opt/

[root@A /]# ls /opt/

[root@A /]# ls /opt/boot/

. :当前目录

[root@A /]# cd /opt

[root@A opt]# cp /etc/passwd . #复制到当前目录下

[root@A opt]# ls

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

**1.强制覆盖**

**命令前面加"反斜杠",临时取消别名**

**[root@A /]# cp -r /boot/ /opt/**

**[root@A /]# \cp -r /boot/ /opt/**

2.多参数的应用,永远会把最后一个参数作为目标,其他的所有参数作为源

[root@A /]# cp -r /home/ /etc/shadow /root /opt/

[root@A /]# ls /opt/

3.复制时,可以重新命名目标路径文档的名字

[root@A /]# rm -rf /opt/\*

[root@A /]# cp /etc/redhat-release /opt/red.txt

[root@A /]# ls /opt/

[root@A /]# cp -r /boot/ /opt/admin

[root@A /]# ls /opt/

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

请描述下列命令操作的意义

[root@A /]# rm -rf /opt/\* #清空/opt目录内容

[root@A /]# cp -r /home /opt/test

#将/home复制到/opt下改名为test

[root@A /]# cp -r /home /opt/test

#将/home复制到/opt/test目录下

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

**2018-07-02当天作业：**

案例：复制、删除、移动

1. 在目录/opt下创建一个子目录 nsd

mkdir /opt/nsd

1. 在目录/opt/nsd/创建文件readme.txt,利用vim写入内容 I Love Linux

vim /opt/nsd/readme.txt/

cat /opt/nsd/readme.txt/

I Love Linux

1. 将/etc/passwd 和 /etc/resolv.conf同时拷贝到/opt/nsd目录下

cp /etc/passwd/ /etc/resolv.conf/ /opt/nsd

1. 将文件 /etc/redhat-release复制到 /root/ 下，同时 改名为 version.txt

cp /etc/redhat-release /root/ version.txt

1. 将文件 /root/version.txt 移动到/opt/nsd/目录下

mv /root/version.txt /opt/nsd/

1. 将/home目录复制到/opt/nsd/目录下

cp -r /home/ /opt/nsd/

案例：ls命令练习

1. 查看根目录下内容

ls /

1. 显示根目录本身详细属性

ls -ld /

1. 切换到/boot目录，显示当前目录内容的详细属性，并加上易读的容量单位

cd /boot/

ls -lh /boot/

1. 显示/root的全部内容，包括隐藏文档

ls -A /root/

1. 显示/bin/bash程序，详细属性

ls -l /bin/bash/

1. 显示/dev/cdrom光驱设备，详细属性

ls -l /dev/cdrom

案例：mount挂载

1. 在根目录下创建目录结构/rhel7/dvd

mkdir -p //rhel7/dvd

2. 利用图形将光盘文件rhel-server-7.4-x86\_64-dvd.iso放入光驱设备中

3. 将光驱设备挂载到/rhel7/dvd目录，以/rhel7/dvd目录作为其访问点

mount /dev/cdrom/ /rhel7/dvd

4. 查看/rhel7/dvd/Packages目录内容

ls /rhel7/dvd/Packages  
 5. 将/rhel7/dvd/Packages目录中以vsftpd开头的软件包，拷贝到/opt下

cp /rhel7/dvd/Packages/vsftpd\*/ /opt/

案例：别名与vim的使用

1. 为虚拟机定义一个别名，执行byebye可以实现关闭系统

alias byebye=’poweroff’

1. 利用vim 修改文件/etc/hostname将其原有内容全部删除，写入新的内容为student.tmooc.cn

vim /etc/hostname

cat /etc/hostname

student.tmooc.cn

**2018-07-03（第四天）**

**早晨默写：**

1.Linux中命令的执行依赖于解释器，默认的解释器是哪个程序？

/bin/bash

2.如何结束当前正在运行的命令，快捷键是？

Ctrl + c

3.切换上一级目录的操作命令？去往用户root的家目录？

cd ..

cd /root 或 cd ~root

4.ls常用选项有哪些？不用写作用

-l -A -h -d

5. 常见的通配符都有哪些？不用写作用

\* ? [] {}

6.如何定义一个别名？如何查看所有的别名？如何删除一个别名？

alias 别名='实际运行的命令'

alias

unalias

7.移动的命令是？

mv

8.强制删除一个文件夹以及其内容的命令是？

rm -rf

9.cp命令在使用时，源是一个目录时需要加什么选项？

-r

10.vim编辑器有几种工作模式？分别是什么？

命令模式 输入模式(插入模式 编辑模式) 末行模式

11.在vim末行模式中，输入什么可以不保存退出？

:q!

12.使用 ls 命令完成以下任务,找出/etc/目录下以re开头.conf结尾的文件

ls /etc/re\*.conf

**2018-07-03**

教学环境介绍

每个学员机上有三台预先配置好的虚拟机

– server —— 作为练习用服务器

– desktop —— 作为练习用客户机

– classroom —— 提供网关/DNS/软件素材/DHCP服务器等资源

**开机顺序:**

**首先 classroom 开启,在开启server与desktop**

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

在真机上操作:

1.先重置资源服务器

[root@room9pc13 ~]# rht-vmctl reset classroom

[root@room9pc13 ~]# rht-vmctl reset server

[root@room9pc13 ~]# rht-vmctl reset desktop

2.进行登陆

利用root 密码redhat 登陆虚拟机server与虚拟机desktop

3.虚拟机基本信息

虚拟机Server

1)查看操作系统版本: RHEL7.0

2)查看eth0的IP地址:172.25.0.11/24

3)查看主机名:server0.example.com

虚拟机desktop

1)查看操作系统版本: RHEL7.0

2)查看eth0的IP地址:172.25.0.10/24

3)查看主机名:desktop0.example.com

4.默认真机可以与, 两台虚拟机Server/虚拟机desktop 进行通信

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

真机远程管理虚拟机

1.前提可以ping通

2.远程管理 ssh 用户名@对方IP地址

[root@room9pc01 ~]# ssh root@172.25.0.11

[root@server0 ~]# hostname

[root@server0 ~]# ifconfig

[root@server0 ~]# exit #退出远程管理回到真机

**补充: ctrl + shift + t 新开一个终端**

**3.远程管理选项 -X(大写) 在本机运行对方的图形程序**

[root@server0 ~]# exit

**[root@room9pc01 ~]# ssh -X root@172.25.0.11**

[root@server0 ~]# firefox

[root@server0 ~]# firewall-config #防火墙的图形工具

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

**制作永久的别名,方便远程管理**

**真机:永久的别名配置文件/root/.bashrc (系统级配置文件)**

**[root@room9pc01 ~]# vim /root/.bashrc**

**alias gos='ssh -X root@172.25.0.11'**

**alias god='ssh -X root@172.25.0.10'**

**[root@room9pc01 ~]# 新开一个全新的终端**

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

软件包的管理

1.关闭虚拟机Server添加虚拟的光驱设备

2.系统显示光盘内容

[root@server0 ~]# mkdir /dvd

[root@server0 ~]# mount /dev/cdrom /dvd/

mount: /dev/sr0 写保护，将以只读方式挂载

[root@server0 ~]# ls /dvd/

[root@server0 ~]# ls /dvd/Packages

• RPM Package Manager,RPM包管理器

– rpm -q 软件名... #查询软件是否安装

– rpm -ivh 软件名-版本信息.rpm... #安装软件包

– rpm -e 软件名... #卸载软件包

[root@server0 ~]# rpm -q firefox firefoxfirefox-24.5.0-1.el7.x86\_64

[root@server0 ~]# rpm -q zip

zip-3.0-10.el7.x86\_64

[root@server0 ~]# rpm -q haha

未安装软件包 haha

[root@server0 ~]# rpm -q net-tools

net-tools-2.0-0.17.20131004git.el7.x86\_64

[root@server0 ~]# rpm -q mysql

未安装软件包 mysql

[root@server0 ~]# rpm -q mariadb

未安装软件包 mariadb

[root@server0 ~]# rpm -q vsftpd #查询是否安装

未安装软件包 vsftpd

# rpm -ivh /dvd/Packages/vsftpd-3.0.2-22.el7.x86\_64.rpm

[root@server0 ~]# rpm -q vsftpd #查询是否安装成功

vsftpd-3.0.2-22.el7.x86\_64

[root@server0 ~]# rpm -e vsftpd #卸载软件程序

[root@server0 ~]# rpm -q vsftpd #查询是否卸载成功

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

软件包依赖关系

错误：依赖检测失败：

bind = 32:9.9.4-50.el7 被 bind-

chroot-32:9.9.4-50.el7.x86\_64 需要

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

Yum软件包仓库,自动解决依赖关系

服务:为客户端安装软件包,并解决依赖关系

**服务端: 1.众多的软件包 2.仓库清单文件 3.共享的服务**

服务端:classroom.example.com搭建Web服务,共享光盘所有内容RHEL7.0

http://classroom.example.com/content/rhel7.0/x86\_64/dvd/

**客户端: 书写配置文件/etc/yum.repos.d/\*.repo**

**正确的文件与错误的文件会相互影响**

[root@server0 ~]# rm -rf /etc/yum.repos.d/\*

[root@server0 ~]# vim /etc/yum.repos.d/rhel7.repo

**[rhel7] #仓库标识(可以随意写)**

**name=rhel7.0 #仓库描述信息(可以随意写)**

**baseurl=http://classroom.example.com/content/rhel7.0/x86\_64/dvd/ #具体服务端访问位置**

**enabled=1 #是否启用该文件**

**gpgcheck=0 #是否检测签名认证**

**[root@server0 ~]# yum repolist #列出仓库信息**

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

**Yum使用**

**yum -y install 软件名**

[root@server0 ~]# yum -y install httpd

[root@server0 ~]# yum -y install gcc

[root@server0 ~]# yum -y install mariadb-server

[root@server0 ~]# yum -y install sssd

[root@server0 ~]# yum -y install system-config-kickstart

[root@server0 ~]# yum -y install xeyes

[root@server0 ~]# xeyes

yum remove 软件名 #卸载

yum search 关键字 #查询

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

升级Linux内核

只使用于虚拟机

http://classroom.example.com/content/rhel7.0/x86\_64/errata/Packages/kernel-3.10.0-123.1.2.el7.x86\_64.rpm

下载软件包

**• 使用wget下载工具,默认下载到当前**

**– wget 软件包的URL网址**

[root@server0 ~]# wget http://classroom.example.com/content/rhel7.0/x86\_64/errata/Packages/kernel-3.10.0-123.1.2.el7.x86\_64.rpm

[root@server0 ~]# uname -r

3.10.0-123.el7.x86\_64

# rpm -ivh kernel-3.10.0-123.1.2.el7.x86\_64.rpm

[root@server0 ~]# uname -r

3.10.0-123.el7.x86\_64

[root@server0 ~]# reboot

[root@room9pc01 ~]# gos

Last login: Tue Jul 3 15:38:14 2018 from 172.25.0.250

[root@server0 ~]# uname -r

3.10.0-123.1.2.el7.x86\_64

[root@server0 ~]#

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

配置网络

**一 配置永久的主机名/etc/hostname**

**[root@server0 ~]# vim /etc/hostname**

A.tedu.cn

[root@server0 ~]# cat /etc/hostname

[root@server0 ~]# exit

[root@room9pc01 ~]# gos

[root@A ~]# hostname

A.tedu.cn

[root@A ~]#

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

二 配置永久 IP地址 子网掩码 网关地址

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

命令配置

nmcli connection

1.查看命令识别的网卡名

[root@A ~]# nmcli connection show

名称 ...............

System eth0 .........

**2.配置IP地址 子网掩码 网关地址**

**[root@A ~]# nmcli connection modify 'System eth0'**

**ipv4.method manual**

**ipv4.addresses '172.25.0.110/24 172.25.0.254'**

**connection.autoconnect yes**

nmcli connection 修改 '网卡识别名称'

ipv4.方法 手工配置

ipv4.地址 ' ip地址/子网掩码 网关地址'

每次开机自动启用网络配置

[root@A ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0

**3.激活配置**

**[root@A ~]# nmcli connection up 'System eth0'**

由于IP地址的变化,所以终端会失联,新开一个终端进行远程管理新的IP地址

[root@room9pc01 ~]# ssh -X root@172.25.0.110

[root@A ~]#

4.网关查看

[root@A ~]# route

.......... Gateway .......... ..........

.......... 172.25.0.254 ..........

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

**三 配置永久DNS地址,永久配置文件/etc/resolv.conf**

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

**nameserver 172.25.254.254**

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

**验证:**

**[root@A ~]# nslookup classroom.example.com**

**[root@A ~]# nslookup server0.example.com**

**[root@A ~]# nslookup desktop0.example.com**

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

**2018-07-03作业**

案例练习,准备:

rht-vmctl reset classroom

rht-vmctl reset server

rht-vmctl reset desktop

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

案例1：真机上操作：别名与vim的使用

1. 定义一个永久别名（修改/root/.bashrc文件），执行c可以实现还原虚拟机classroom

vim /root/.bashrc

head -2 /root/.bashrc

# .bashrc

alias c='rht-vmctl reset classroom'

2. 定义一个永久别名（修改/root/.bashrc文件），执行s可以实现还原虚拟机server

vim /root/.bashrc

head -3 /root/.bashrc

# .bashrc

alias c='rht-vmctl reset classroom'

alias s='rht-vmctl reset server'

3. 定义一个永久别名（修改/root/.bashrc文件），执行d可以实现还原虚拟机Desktop

vim /root/.bashrc

head -4 /root/.bashrc

# .bashrc

alias c='rht-vmctl reset classroom'

alias s='rht-vmctl reset server'

alias d='rht-vmctl reset desktop'

案例2：虚拟机Server上操作：复制、删除、移动及vim文本编辑器

1. 在目录/mnt下创建一个子目录public

mkdir /mnt/public

2. 在目录/mnt/public 创建文件linux.txt,利用vim写入内容 Study Linux

vim /mnt/public/linux.txt

cat /mnt/public/linux.txt

Study Linux

3. 将/mnt/public/linux.txt文件复制到/root目录下，同时 改名为 study.txt

cp /mnt/public/linux.txt /root/study.txt

4. 利用vim 修改文件/etc/hostname将其原有内容全部删除，写入新的内容为www.qq.com

vim /etc/hostname

cat /etc/hostname

www.qq.com

5. 将/etc/passwd 、/etc/resolv.conf、/etc/hostname 同时拷贝到/mnt/public/目录下

cp /etc/passwd /etc/resolv.conf /etc/hostname /mnt/public/

6. 将文件 /mnt/public/hostname 重改名为 stu.txt

mv /mnt/public/hostname /mnt/public/stu.txt

7. 创建目录结构/mnt/public/test/vm

mkdir -p /mnt/public/test/vm

7. 将目录 /boot内容中以 vm 开头的 复制到/mnt/public/test/vm目录下

cp /boot/vm\* /mnt/public/test/vm

8. 将/home目录复制到/mnt/public/test/目录下

cp -r /home /mnt/public/test

案例3：虚拟机Server上操作：ls与cat、head

1. 显示根目录下所有内容

ls /

2. 显示/etc目录下所有已tab结尾的文件

ls /etc/\*tab

3. 显示/etc/resolv.conf文件的详细属性并加上易读的单位

ls -lh /etc/resolv.conf

4. 显示/etc/sysconfig/network-scripts/ifcfg-eth0文件的详细属性并加上易读的单位

ls -lh /etc/sysconfig/network-scripts/ifcfg-eth0

5. 显示/etc/sysconfig/network-scripts/ifcfg-eth0文件的头4行内容

head -4 /etc/sysconfig/network-scripts/ifcfg-eth0

案例4：虚拟机Server上操作：搭建Yum仓库

为 server0 指定可用的 yum 软件源

– YUM软件库的地址为 http://classroom.example.com/content/rhel7.0/x86\_64/dvd/

– 将此配置为虚拟机 server0 的默认软件仓库

– 确认可用的仓库列表

– 利用yum仓库安装sssd软件

– 利用yum仓库安装xeyes软件

rm -rf /etc/yum.repos.d/\*

vim /etc/yum.repos.d/rhel7.repo

cat /etc/yum.repos.d/rhel7.repo

[rhel7]

name=rhel7.0

baseurl=http://classroom.example.com/content/rhel7.0/x86\_64/dvd/

enabled=1

gpgcheck=0

yum repolist

yum -y install sssd

yum -y install xeyes

案例5:为虚拟机 server 配置以下静态地址参数

– 主机名:server0.example.com

– IP地址:172.25.0.100

– 子网掩码:255.255.255.0

– 默认网关:172.25.0.254

– DNS服务器:172.25.254.254

vim /etc/hostname

cat /etc/hostname

server0.example.com

nmcli connection modify 'System eth0' ipv4.method manual ipv4.addresses '172.25.0.100/24 172.25.0.254' connection.autoconnect yes

nmcli connection up 'System eth0'

vim /etc/resolv.conf

cat /etc/resolv.conf

nameserver 172.25.254.254

**2018-07-04（第五天）**

教学环境介绍

[root@room9pc01 ~]# rht-vmctl reset classroom

classroom [OK]

域 classroom 已开始

[root@room9pc01 ~]# rht-vmctl reset server

server [OK]

域 server 已开始

[root@room9pc01 ~]# gos

Last login: Fri Jan 12 18:54:14 2018 from 172.25.0.250

/usr/bin/xauth: file /root/.Xauthority does not exist

[root@server0 ~]#

[root@server0 ~]# yum clean all #清空Yum缓存

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

重定向输出操作

>:覆盖重定向

>>:追加重定向

[root@server0 ~]# ls --help > /opt/ls.txt

[root@server0 ~]# ls /opt/

[root@server0 ~]# less /opt/ls.txt

[root@server0 ~]# hostname

[root@server0 ~]# hostname > /opt/ls.txt

[root@server0 ~]# cat /opt/ls.txt

[root@server0 ~]# cat /etc/redhat-release

[root@server0 ~]# cat /etc/redhat-release > /opt/ls.txt

[root@server0 ~]# cat /opt/ls.txt

[root@server0 ~]# hostname >> /opt/ls.txt

[root@server0 ~]# cat /opt/ls.txt

[root@server0 ~]# hostname >> /opt/ls.txt

[root@server0 ~]# cat /opt/ls.txt

[root@server0 ~]# echo 123456789

123456789

[root@server0 ~]# echo 123456789 >> /opt/ls.txt

[root@server0 ~]# cat /opt/ls.txt

[root@server0 ~]# echo A.tedu.cn > /etc/hostname

[root@server0 ~]# cat /etc/hostname

# echo nameserver 172.25.254.254 > /etc/resolv.conf

# cat /etc/resolv.conf

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

| 管道操作: 将前面命令的输出结果,作为参数交由后面命令处理

显示/etc/passwd内容8到12行?

[root@server0 ~]# head -12 /etc/passwd

[root@server0 ~]# head -12 /etc/passwd | tail -5

[root@server0 ~]# cat -n /etc/passwd | head -12 | tail-5

[root@server0 ~]# ifconfig | head -2

[root@server0 ~]# ls --help | less

[root@server0 ~]# ls --help | cat -n | less

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

管理用户和组

用户帐号: 1.可以登陆操作系统 2.访问控制(不同用户具备不同权限)

组帐号 : 方便管理用户

唯一标识: UID GID

管理员的UID永远为: UID 0

普通用户的UID默认从: UID 1000 起始

**组: 基本组(私有组) 附加组(从属组)**

**Linux用户至少属于一个组**

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

添加用户

用户基本信息存放在 /etc/passwd 文件

[root@A ~]# head -1 /etc/passwd

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

**用户名:密码占位符:UID:基本组的GID:用户描述信息:家目录:解释器程序**

**• 使用 useradd 命令**

**– useradd [选项]... 用户名**

**• 常用命令选项**

**– -u 用户id、-d 家目录路径、-s 解释器程序、-G 附加组**

[root@A ~]# useradd nsd01

[root@A ~]# grep nsd01 /etc/passwd

[root@A ~]# id nsd01 #查看用户基本信息

[root@A ~]# useradd -u 1600 nsd06 #指定UID创建用户

[root@A ~]# grep nsd06 /etc/passwd

[root@A ~]# useradd -d /opt/nsd07 nsd07 #指定家目录创建用户

[root@A ~]# grep nsd07 /etc/passwd

[root@A ~]# ls /opt/

**[root@A ~]# groupadd tarena #创建组**

**[root@A ~]# useradd -G tarena nsd08 #指定附加组创建用户**

[root@A ~]# id nsd08

[root@A ~]# useradd -G tarena nsd09

[root@A ~]# id nsd09

**/sbin/nologin : 禁止用户登陆操作系统**

[root@A ~]# useradd -s /sbin/nologin nsd10

[root@A ~]# grep nsd10 /etc/passwd

[root@A ~]# useradd -s /sbin/nologin nsd11

[root@A ~]# grep nsd11 /etc/passwd

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

设置登录密码

用户密码信息存放在 /etc/shadow 文件

[root@A ~]# head -1 /etc/shadow

用户名:密码加密之后的字符串:上一次修改密码的时间 ........

**• 使用 passwd 命令**

**– passwd [用户名]**

**非交互式设置密码**

**– echo '密码' | passwd --stdin 用户名**

[root@A ~]# passwd nsd01

[root@A ~]# echo 123 | passwd --stdin nsd01

[root@A ~]# echo redhat | passwd --stdin nsd01

[root@A ~]# echo 123456 | passwd --stdin nsd01

命令行临时切换身份: su - 用户名

successfully.(成功)

**[root@A ~]# echo 123 | passwd --stdin nsd01**

[root@A ~]# su - nsd01 #临时切换用户身份为nsd01

[nsd01@A ~]$ passwd

Changing password for user nsd01.

Changing password for nsd01.

(current) UNIX password: #输入旧密码

New password: #输入新密码

Retype new password: #输入新密码

passwd: all authentication tokens updated successfully.

[nsd01@A ~]$ exit

logout

[root@A ~]#

**修改用户属性**

**• 使用 usermod 命令**

**– usermod [选项]... 用户名**

**• 常用命令选项**

**– -u 用户id、-d 家目录路径、-s 登录Shell、-G 附加组**

[root@A ~]# useradd harry

[root@A ~]# grep harry /etc/passwd

[root@A ~]# id harry

# usermod -u 1700 -d /opt/harry -s /sbin/nologin -G tarena harry

[root@A ~]# id harry

[root@A ~]# grep harry /etc/passwd

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

**删除用户**

**• 使用 userdel 命令**

**– userdel [-r] 用户名 #连同家目录一并删除**

[root@A ~]# userdel nsd01

[root@A ~]# id nsd01

id: nsd01: no such user

[root@A ~]# userdel -r nsd02 #连同家目录一并删除

[root@A ~]# ls /home/

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

**管理组账号**

**添加组**

**组基本信息存放在 /etc/group 文件**

**[root@A ~]# grep stugrp /etc/group**

**stugrp:x:1605:**

**组名:组的密码占位符:GID:组成员列表**

**• 使用 groupadd 命令**

**– groupadd [-g 组ID] 组名**

[root@A ~]# groupadd stugrp

[root@A ~]# grep stugrp /etc/group

stugrp:x:1605:

[root@A ~]# useradd natasha

[root@A ~]# useradd kenji

[root@A ~]# useradd jack

[root@A ~]# useradd kaka

**• 使用 gpasswd 命令**

**– gpasswd -a 用户名 组名**

**– gpasswd -d 用户名 组名**

[root@A ~]# gpasswd -a natasha stugrp #添加用户到组中

[root@A ~]# grep stugrp /etc/group

[root@A ~]# gpasswd -a kenji stugrp

[root@A ~]# grep stugrp /etc/group

[root@A ~]# gpasswd -a jack stugrp

[root@A ~]# grep stugrp /etc/group

[root@A ~]# gpasswd -d kenji stugrp #将用户从组中删除

[root@A ~]# grep stugrp /etc/group

[root@A ~]# gpasswd -a kaka stugrp

[root@A ~]# grep stugrp /etc/group

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

删除组

**• 使用 groupdel 命令**

**– groupdel 组名**

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

tar备份与恢复,打包及压缩

1.方便对零散文档的管理 2.减小空间的占用

Linux独有的压缩的格式

gzip------->.gz

bzip2------>.bz2

xz--------->.xz

打包:

tar [选项] /路径/压缩包的名字 /被归档压缩的源文件 ......

解包:

tar [选项] /路径/压缩包的名字 -C(大写) 释放位置

**• tar 集成备份工具**

**– -c:创建归档**

**– -x:释放归档**

**– -f:指定归档文件名称**

**– -z、-j、-J:调用 .gz、.bz2、.xz 格式的工具进行处理**

**– -t:显示归档中的文件清单**

**– -C(大写): 释放位置**

# tar -zcf /opt/test01.tar.gz /home/ /etc/passwd

# tar -jcf /opt/file.tar.bz2 /home/ /etc/passwd

# tar -Jcf /opt/nsd.tar.xz /home/ /etc/passwd

# ls /opt/

# mkdir /file

# tar -xf /opt/test01.tar.gz -C /file/

# ls /file/

# ls /file/home/

# ls /file/etc/

# tar -zcf /opt/back.tar.gz /etc/passwd /etc/shadow /boot

# ls /opt/

# mkdir /user

# tar -xf /opt/back.tar.gz -C /user

# ls /user

使用 tar 工具完成以下备份任务:

– 创建一个名为 /root/backup.tar.bz2 的归档文件

– 其中包含 /usr/local 目录中的内容

– tar 归档必须使用 bzip2 进行压缩

[root@A ~]# tar -jcf /root/backup.tar.bz2 /usr/local/

[root@A ~]# ls /root/

[root@A ~]# tar -tf /root/backup.tar.bz2 #查看包里面的内容

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

NTP时间同步

服务: NTP服务器为客户机提供标准时间

– NTP客户机需要与NTP服务器保持沟通

服务端: classroom.example.com

客户端:虚拟机Server0

1.安装软件chrony,与服务端沟通时间的软件

[root@A ~]# yum -y install chrony

已加载插件：langpacks

rhel7 | 4.1 kB 00:00

软件包 chrony-1.29.1-1.el7.x86\_64 已安装并且是最新版本

无须任何处理

[root@A ~]# rpm -q chrony

**2.修改主配置文件 /etc/chrony.conf**

[root@A ~]# vim /etc/chrony.conf

以#开头的行,为注释行

#server 0.rhel.pool.ntp.org iburst #前面加'#'变成注释

#server 1.rhel.pool.ntp.org iburst #前面加'#'变成注释

#server 2.rhel.pool.ntp.org iburst #前面加'#'变成注释

server classroom.example.com iburst

**3.重起chronyd服务**

**[root@A ~]# systemctl restart chronyd #重起服务**

**[root@A ~]# systemctl enable chronyd #设置服务随机自起动**

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

验证:

1.查看系统时间

[root@A ~]# date

[root@A ~]# date -s "年-月-日 时:分:秒"

[root@A ~]# date -s "2000-10-1 12:50"

[root@A ~]# date

[root@A ~]# systemctl restart chronyd

[root@A ~]# date

[root@A ~]# date

**配置文件 /etc/chrony.conf**

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

cron计划任务

• 用途:按照设置的时间间隔为用户反复执行某一项固定的系统任务

• 软件包:cronie、crontabs

• 系统服务:crond

• 日志文件:/var/log/crond

如何编写crontab任务记录

**分 时 日 月 周 任务命令行**

**\* \* \* \* \***

**\*:匹配范围内任意时间**

**,:分隔多个不连续的时间点**

**-:指定连续时间范围**

**/n:指定时间频率,每n ..**

**• 使用 crontab 命令**

**– 编辑:crontab -e [-u 用户名]**

**– 查看:crontab -l [-u 用户名]**

– 清除:crontab -r [-u 用户名]

每分钟记录系统时间,将系统时间写入到/opt/time.txt

[root@A ~]# date >> /opt/time.txt

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

[root@A ~]# crontab -e -u root #书写计划任务

\* \* \* \* \* date >> /opt/time.txt

[root@A ~]# crontab -l -u root #查看root计划任务

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

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

**2018-07-04当天作业：**

案例练习,准备:

rht-vmctl reset classroom

rht-vmctl reset server

rht-vmctl reset desktop

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

案例1:为虚拟机 server 配置以下静态地址参数

– 主机名:server0.example.com

– IP地址:172.25.0.120

– 子网掩码:255.255.255.0

– 默认网关:172.25.0.254

– DNS服务器:172.25.254.254

[root@server0 ~]# vim /etc/hostname

[root@server0 ~]# cat /etc/hostname

server0.example.com

[root@server0 ~]# nmcli connection modify 'System eth0' ipv4.method manual ipv4.addresses '172.25.0.120/24 172.25.0.254' connection.autoconnect yes

[root@server0 ~]# nmcli connection up 'System eth0'

[root@server0 ~]# vim /etc/resolv.conf

[root@server0 ~]# cat /etc/resolv.conf

nameserver 172.25.0.254

案例2:指定yum软件源

为 server0 指定可用的 yum 软件源

– YUM软件库的地址为 http://classroom.example.com/content/rhel7.0/x86\_64/dvd

– 将此配置为虚拟机 server0 的默认软件仓库

– 确认可用的仓库列表

– 利用yum仓库安装system-config-kickstart

[root@server0 ~]# rm -rf /etc/yum.repos.d/\*

[root@server0 ~]# vim /etc/yum.repos.d/rhel7.repo

[root@server0 ~]# cat /etc/yum.repos.d/rhel7.repo

[rhel7]

name=rhel7.0

baseurl=http://classroom.example.com/content/rhel7.0/x86\_64/dvd

enabeld=1

gpgcheck=0

[root@server0 ~]# yum repolist

[root@server0 ~]# yum -y install system-config-kickstart

案例3:grep基本用法

1）将/etc/passwd中包含root的行写入文件/opt/root.txt

[root@server0 ~]# grep root /etc/passwd > /opt/root.txt

2）将/etc/passwd中包含nologin的行追加写入文件/opt/root.txt

[root@server0 ~]# grep nologin /etc/passwd >> /opt/root.txt

案例4:tar制作/释放归档压缩包(zcf、ztf、zxf、jcf、jtf、jxf、cf、tf）

首先创建/root/boothome/与/root/usrsbin/目录

mkdir /root/boothome /root/usrsbin

1）备份/boot、/home这两个文件夹，保存为boothome.tar.gz文件

[root@server0 ~]# tar -zcf /root/boothome/boothome.tar.gz /boot /home

2）查看boothome.tar.gz文件内包含哪些内容

[root@server0 ~]# tar -tf /root/boothome/boothome.tar.gz

3）将boothome.tar.gz释放到文件夹/root/boothome/下

[root@server0 ~]# tar -xf /root/boothome/boothome.tar.gz -C /root/boothome

4）备份/usr/sbin目录，保存为usrsbin.tar.bz2文件

[root@server0 ~]# tar -jcf /root/usrsbin/usrubin.tar.bz2 /usr/sbin

5）查看usrsbin.tar.bz2文件内包含哪些内容

[root@server0 ~]# tar -tf /root/boothome/boothome.tar.gz

6）将usrsbin.tar.bz2释放到/root/usrsbin/文件夹下

[root@server0 ~]# tar -xf /root/usrsbin/usrubin.tar.bz2 -C /root/usrsbin/

案例5：添加并测试用户账号

1）创建一个名为stu01的用户账号

[root@server0 ~]# useradd stu01

2）检查/etc/passwd、/etc/shadow文件的最后一行

[root@server0 ~]# tail -1 /etc/passwd

[root@server0 ~]# tail -1 /etc/shadow

3）检查/home/新增加的宿主目录（家目录）

[root@server0 ~]# ls /home

4）为用户stu01设置一个密码（123456）

[root@server0 ~]# echo 123456 | passwd --stdin stu01

案例6：添加账号时设置不同属性

1）新建用户nsd01，宿主目录位于/opt/nsd01

[root@server0 ~]# useradd -d /opt/nsd01 nsd01

2）新建系统账号sys01，将UID设为1234，登录Shell设为/sbin/nologin

[root@server0 ~]# useradd -u 1234 -s /sbin/nologin sys01

3）为用户sys01设置密码，并测试是否能够登录

[root@server0 ~]# echo 123456 | passwd --stdin sys01

4）新建用户admin，将其基本组设为users，附加组设为adm、root

案例7：passwd设置密码

1）给用户nsd01设置密码123456

[root@server0 ~]# passwd nsd01

更改用户 nsd01 的密码 。

新的 密码：

无效的密码： 密码少于 8 个字符

重新输入新的 密码：

passwd：所有的身份验证令牌已经成功更新。

2）采用--stdin方式将用户nsd01的密码设为654321

[root@server0 ~]# echo 654321 |passwd --stdin nsd01

更改用户 nsd01 的密码 。

passwd：所有的身份验证令牌已经成功更新。

案例8：usermod修改用户

1）新建一个用户nsd03，将宿主目录设为/opt/home03，并设置密码

[root@server0 ~]# useradd -d /opt/home03 nsd03

[root@server0 ~]# echo 123456 |passwd --stdin nsd03

更改用户 nsd03 的密码 。

passwd：所有的身份验证令牌已经成功更新。

2）设置nsd03密码为redhat

[root@server0 ~]# echo redhat |passwd --stdin nsd03

更改用户 nsd03 的密码 。

passwd：所有的身份验证令牌已经成功更新。

3）将用户nsd03的宿主目录改为/home/nsd03

[root@server0 ~]# usermod -d /home/nsd03 nsd03

4）将用户sys01的登录Shell改为/bin/bash

[root@server0 ~]# usermod -s /bin/bash sys01

案例9：组账号基本管理

1）新建组账号stugrp

[root@server0 ~]# groupadd stugrp

2）为stugrp组添加三个成员用户（user01、root、zhangsan）

[root@server0 ~]# useradd user01

[root@server0 ~]# useradd zhangsan

[root@server0 ~]# gpasswd -a user01 stugrp

正在将用户“user01”加入到“stugrp”组中

[root@server0 ~]# gpasswd -a root stugrp

正在将用户“root”加入到“stugrp”组中

[root@server0 ~]# gpasswd -a zhangsan stugrp

正在将用户“zhangsan”加入到“stugrp”组中

3）从stugrp组删除一个成员（user01）

[root@server0 ~]# gpasswd -d user01 stugrp

正在将用户“user01”从“stugrp”组中删除

案例10：配置NTP网络时间客户端

配置虚拟机 server0，自动校对系统时间

NTP服务器位于 classroom.example.com

此客户机的时间与NTP服务器的时间保持同步

[root@server0 ~]# yum -y install chrony

[root@server0 ~]# vim /etc/chrony.conf

[root@server0 ~]# head -6 /etc/chrony.conf

# Use public servers from the pool.ntp.org project.

# Please consider joining the pool (http://www.pool.ntp.org/join.html).

#server 0.rhel.pool.ntp.org iburst

#server 1.rhel.pool.ntp.org iburst

#server 2.rhel.pool.ntp.org iburst

server classroom.example.com iburst

[root@server0 ~]# systemctl restart chronyd

[root@server0 ~]# systemctl enable chronyd

**2018-07-05（第六天）**

教学环境介绍

[root@room9pc01 ~]# rht-vmctl reset classroom

classroom [OK]

域 classroom 已开始

[root@room9pc01 ~]# rht-vmctl reset server

server [OK]

域 server 已开始

[root@room9pc01 ~]# gos

Last login: Fri Jan 12 18:54:14 2018 from 172.25.0.250

/usr/bin/xauth: file /root/.Xauthority does not exist

[root@server0 ~]#

[root@server0 ~]# yum clean all #清空Yum缓存

[root@server0 ~]# yum repolist

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

基本权限

基本权限的类别

• 访问方式(权限)

– 读取:允许查看内容 -read r

– 写入:允许修改内容 -write w

– 可执行:允许运行和切换 -execute x

对于文本文件:

r(读取): cat less head tail grep

w(写入): vim 保存并退出

x(可执行): Shell脚本

• 权限适用对象(归属)

– 所有者:拥有此文件/目录的用户-user u

– 所属组:拥有此文件/目录的组-group g

– 其他用户:除所有者、所属组以外的用户-other o

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

查看权限

• 使用 ls -l 命令

– ls -ld 文件或目录...

权限位 硬连接数 属主 属组 大小 最后修改时间 文件/目录名称

**以 - 开头:文本文件**

**以d 开头:目录**

**以l 开头:快捷方式**

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

**设置基本权限**

**• 使用 chmod 命令**

– chmod [-R] 归属关系+-=权限类别 文档..

[-R] 递归设置

[root@server0 ~]# mkdir /nsd01

[root@server0 ~]# ls -ld /nsd01

[root@server0 ~]# chmod u-w /nsd01

[root@server0 ~]# ls -ld /nsd01

[root@server0 ~]# chmod g+w /nsd01

[root@server0 ~]# ls -ld /nsd01

[root@server0 ~]# chmod u=rwx,g=rx,o=rx /nsd01

[root@server0 ~]# ls -ld /nsd01

[root@server0 ~]# chmod ugo=rwx /nsd01

[root@server0 ~]# ls -ld /nsd01

[root@server0 ~]# chmod ugo=--- /nsd01

[root@server0 ~]# ls -ld /nsd01

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

**如何判断用户具备的权限:**

**1.判断用户对于文档的角色 所有者>所属组>其他人 匹配及停止**

**2.查看相应角色权限位置**

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

Permission denied : 权限不足

**目录的 r 权限:能够 ls 浏览此目录内容**

**目录的 w 权限:能够执行 rm/mv/cp/mkdir/touch/ 等更改目录内容的操作**

**目录的 x 权限:能够 cd 切换到此目录**

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

以root用户新建/nsddir/目录，在此目录下新建readme.txt文件，

并进一步完成下列操作

1）使用户lisi能够在此目录下创建子目录 切换用户 su - lisi

chmod o+w /nsddir/

2）使用户lisi不能够在此目录下创建子目录

chmod o-w /nsddir/

3）使用户lisi能够修改readme.txt文件

chmod o+w /nsddir/readme.txt

4）调整此目录的权限，使所有用户都不能cd进入此目录

chmod u-x,g-x,o-x /nsddir/

5）为此目录及其下所有文档设置权限 rwxr-x---

chmod -R u=rwx,g=rx,o=--- /nsddir/

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

**设置文档归属**

**– chown [-R] 属主 文档...**

**– chown [-R] :属组 文档...**

**– chown [-R] 属主:属组 文档..**

[root@server0 /]# mkdir /nsd03

[root@server0 /]# ls -ld /nsd03

[root@server0 /]# groupadd tarena

[root@server0 /]# chown lisi:tarena /nsd03

[root@server0 /]# ls -ld /nsd03

[root@server0 /]# chown zhangsan /nsd03 #修改所有者

[root@server0 /]# ls -ld /nsd03

[root@server0 /]# chown :root /nsd03 #修改所属组

[root@server0 /]# ls -ld /nsd03

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

利用root用户新建/nsd06目录，并进一步完成下列操作

1）将属主设为gelin01，属组设为tarena组

[root@server0 /]# useradd gelin01

[root@server0 /]# groupadd tarena

[root@server0 /]# chown gelin01:tarena /nsd06

2）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@server0 /]# chmod o=--- /nsd06

3）使用户gelin02能进入、查看此目录

[root@server0 /]# useradd gelin02

[root@server0 /]# gpasswd -a gelin02 tarena

4）将gelin01加入tarena组, 将nsd06目录的权限设为rw-r-x---

再测试gelin01用户能否进入此目录

[root@server0 /]# chmod u=rw,g=rx /nsd06

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

**附加权限(特殊权限)**

**Set GID**

**• 附加在属组的 x 位上**

**– 属组的权限标识会变为 s**

**– 适用于目录,Set GID可以使目录下新增的文档自动设置与父目录相同的属组**

**– 继承父目录的所属组身份**

**-R:已建立的所有文档**

**SGID:未来建立的**

[root@server0 /]# mkdir /nsd08

[root@server0 /]# chown :tarena /nsd08

[root@server0 /]# ls -ld /nsd08

[root@server0 /]# mkdir /nsd08/abc01

[root@server0 /]# ls -ld /nsd08/abc01 #所属组为root组

[root@server0 /]# chmod g+s /nsd08

[root@server0 /]# ls -ld /nsd08

[root@server0 /]# mkdir /nsd08/abc02

[root@server0 /]# ls -ld /nsd08/abc02 #可以看到所属组变化

[root@server0 /]# touch /nsd08/1.txt

[root@server0 /]# ls -l /nsd08/1.txt #可以看到所属组变化

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

acl访问控制列表

– 大多数挂载的EXT3/4、XFS文件系统默认已支持

• 文档归属的局限性

– 任何人只属于三种角色:属主、属组、其他人

– 无法实现更精细的控制

• **acl访问策略**

**– 能够对个别用户、个别组设置独立的权限**

[root@server0 /]# mkdir /nsd11

[root@server0 /]# chmod o=--- /nsd11

[root@server0 /]# ls -ld /nsd11

[root@server0 /]# su - lisi

[lisi@server0 ~]$ cd /nsd11

-bash: cd: /nsd11: Permission denied

[lisi@server0 ~]$ exit

logout

**[root@server0 /]# setfacl -m u:lisi:rx /nsd11 #为用户单独设置**

[root@server0 /]# getfacl /nsd11 #查看ACL权限

[root@server0 /]# su - lisi

[lisi@server0 ~]$ cd /nsd11

[lisi@server0 nsd11]$ pwd

[lisi@server0 nsd11]$ exit

logout

[root@server0 /]#

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

[root@server0 /]# mkdir /public

[root@server0 /]# chmod ugo=rwx /public

[root@server0 /]# ls -ld /public

drwxrwxrwx. 2 root root 6 7月 5 16:09 /public

[root@server0 /]# id tom

uid=1003(tom) gid=1004(tom) 组=1004(tom)

[root@server0 /]# setfacl -m u:tom:--- /public/

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

**– setfacl [-R] -m u:用户名:权限类别 文档...**

**– setfacl [-R] -m g:组名:权限类别 文档...**

**– setfacl [-R] -b 文档... #清空所有ACL**

**– setfacl [-R] -x u:用户名 文档... #删除指定ACL**

[root@server0 /]# mkdir /nsd12

[root@server0 /]# setfacl -m u:lisi:rwx /nsd12

[root@server0 /]# setfacl -m u:zhangsan:rx /nsd12

[root@server0 /]# setfacl -m u:gelin01:rwx /nsd12

[root@server0 /]# setfacl -m u:gelin02:rx /nsd12

[root@server0 /]# getfacl /nsd12

**[root@server0 /]# setfacl -x u:gelin01 /nsd12 #删除指定用户**

[root@server0 /]# getfacl /nsd12

**[root@server0 /]# setfacl -b /nsd12 #清除所有ACL**

[root@server0 /]# getfacl /nsd12

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

使用LDAP认证

LDAP服务器: 网络用户认证

• 轻量级目录访问协议

– Lightweight Directory Access Protocol

– 由服务器来集中存储并向客户端提供的信息,存储方式类似于DNS分层结构

– 提供的信息包括:用户名、密码、通信录、主机名映 射记录、......

网络用户: 由LDAP服务器提供用户名 密码信息

本地用户: /etc/passwd /etc/shadow

一 搭建LDAP服务器 classroom.example.com

二 搭建客户端,虚拟机Server0

**1.安装软件sssd,与LDAP服务器沟通**

**[root@server0 /]# yum -y install sssd**

**2.安装图形软件配置sssd软件**

**[root@server0 /]# yum -y install authconfig-gtk**

**3.运行图形软件进行配置**

**[root@server0 /]# authconfig-gtk**

选择LDAP

dc=example,dc=com #指定服务端域名

classroom.example.com #指定服务端主机名

勾选TLS加密

使用证书加密: http://classroom.example.com/pub/example-ca.crt

选择LDAP密码

**4.重起服务**

**[root@server0 ~]# systemctl restart sssd**

**[root@server0 ~]# systemctl enable sssd**

**5.验证:**

**[root@server0 ~]# grep ldapuser0 /etc/passwd**

**[root@server0 ~]# id ldapuser0**  #可以看到用户基本信息

**2018-07-05（第六天）作业：**

案例练习,准备:

rht-vmctl reset classroom

rht-vmctl reset server

rht-vmctl reset desktop

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

案例1:指定yum软件源

为 server0 指定可用的 yum 软件源

– YUM软件库的地址为 http://classroom.example.com/content/rhel7.0/x86\_64/dvd/

– 将此配置为虚拟机 server0 的默认软件仓库

– 确认可用的仓库列表

– 利用yum仓库安装xeyes

[root@server0 ~]# rm -rf /etc/yum.repos.d/\*

[root@server0 ~]# vim /etc/yum.repos.d/rhel7.repo

[root@server0 ~]# cat /etc/yum.repos.d/rhel7.repo

[rhel7]

name=rhel7.0

baseurl=http://classroom.example.com/content/rhel7.0/x86\_64/dvd/

enabled=1

gpgcheck=0

[root@server0 ~]# yum repolist

已加载插件：langpacks

rhel7 | 4.1 kB 00:00

(1/2): rhel7/group\_gz | 134 kB 00:00

(2/2): rhel7/primary\_db | 3.4 MB 00:00

源标识 源名称 状态

rhel7 rhel7.0 4,305

repolist: 4,305

[root@server0 ~]# yum -y install xeyes

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

[root@server0 ~]# mkdir /nsddir

[root@server0 ~]# touch /nsddir/readme.txt

2）使用户zhangsan能够在/nsddir/目录下创建/删除子目录

[root@server0 ~]# useradd zhangsan

[root@server0 ~]# id zhangsan

uid=1001(zhangsan) gid=1001(zhangsan) 组=1001(zhangsan)

[root@server0 ~]# ls -ld /nsddir

drwxr-xr-x. 2 root root 23 7月 5 18:58 /nsddir/

[root@server0 ~]# chmod o+w /nsddir/

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

[root@server0 ~]# ls -ld /nsddir/readme.txt

-rw-r--r--. 1 root root 0 7月 5 18:58 /nsddir/readme.txt

[root@server0 ~]# chmod o+w /nsddir/readme.txt

案例3：chown归属设置

1）新建/tarena1目录

[root@server0 ~]# mkdir /tarena1

a）将属主设为gelin01，属组设为tarena组

[root@server0 ~]# useradd gelin01

[root@server0 ~]# groupadd tarena

[root@server0 ~]# chown gelin01:tarena /tarena1

[root@server0 ~]# ls -ld /tarena1

drwxr-xr-x. 2 gelin01 tarena 6 7月 5 19:04 /tarena1

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@server0 ~]# chmod o=--- /tarena1

[root@server0 ~]# ls -ld /tarena1

drwxr-x---. 2 gelin01 tarena 6 7月 5 19:04 /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@server0 ~]# useradd gelin02

[root@server0 ~]# gpasswd -a gelin02 tarena

正在将用户“gelin02”加入到“tarena”组中

3）新建/tarena2目录

[root@server0 ~]# mkdir /tarena2

a）将属组设为tarena

[root@server0 ~]# chown :tarena /tarena2

b）使tarena组的任何用户都能在此目录下创建、删除文件

[root@server0 ~]# chmod g+w /tarena2

[root@server0 ~]# ls -ld /tarena2

drwxrwxr-x. 2 root tarena 6 7月 5 19:14 /tarena2

4）新建/tarena/public目录

[root@server0 ~]# mkdir -p /tarena/public

a）使任何用户对此目录都有rwx权限

[root@server0 ~]# chmod ugo=rwx /tarena/public

b）拒绝zhangsan进入此目录，对此目录无任何权限

[root@server0 ~]# setfacl -m u:zhangsan:--- /tarena/public

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。

[root@server0 ~]# mkdir -p /data/test

[root@server0 ~]# chmod ug=rwx,o=--- /data/test

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

[root@server0 ~]# chown -R zhangsan:tarena /data/test

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。

[root@server0 ~]# chmod g+s /data/test

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

[root@server0 ~]# useradd lisi

[root@server0 ~]# setfacl -m u:lisi:r /etc/shadow

案例5：绑定到LDAP验证服务

– 使用系统 classroom.example.com 提供的LDAP服务

– 验证服务的基本DN是：dc=example,dc=com

– 账户信息和验证信息都是由 LDAP 提供的

– 连接要使用证书加密，证书可以在下面的链接下载：

http://classroom.example.com/pub/example-ca.crt

– 确认本地可以识别ldapuser0用户

[root@server0 ~]# yum -y install sssd

[root@server0 ~]# yum -y install authconfig-gtk

[root@server0 ~]# authconfig-gtk

用户帐号数据库：LDAP

LDAP搜索基础：dc=example,dc=com

LDAP服务器：classroom.example.com

勾选用TLS加密连接，下载CA证书链接：http://classroom.example.com/pub/example-ca.crt

认证方法：LDAP密码

应用

[root@server0 ~]# systemctl restart sssd

[root@server0 ~]# systemctl enable sssd

[root@server0 ~]# grep ldapuser0 /etc/passwd

[root@server0 ~]# id ldapuser0

uid=1700(ldapuser0) gid=1700(ldapuser0) 组=1700(ldapuser0)

[root@server0 ~]# grep ldapuser20 /etc/passwd

[root@server0 ~]# id ldapuser20

uid=1720(ldapuser20) gid=1720(ldapuser20) 组=1720(ldapuser20)

**2018-07-06（第七天）：**

教学环境介绍

[root@room9pc01 ~]# rht-vmctl reset classroom

classroom [OK]

域 classroom 已开始

[root@room9pc01 ~]# rht-vmctl reset server

server [OK]

域 server 已开始

[root@room9pc01 ~]# gos

Last login: Fri Jan 12 18:54:14 2018 from 172.25.0.250

/usr/bin/xauth: file /root/.Xauthority does not exist

[root@server0 ~]#

[root@server0 ~]# yum clean all #清空Yum缓存

[root@server0 ~]# yum repolist

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

**使用LDAP认证**

**LDAP服务器: 网络用户认证**

**• 轻量级目录访问协议**

**– Lightweight Directory Access Protocol**

**– 由服务器来集中存储并向客户端提供的信息,存储方式类似于DNS分层结构**

**– 提供的信息包括:用户名、密码、通信录、主机名映 射记录、......**

**网络用户: 由LDAP服务器提供用户名 密码信息**

**本地用户: /etc/passwd /etc/shadow**

**一 搭建LDAP服务器 classroom.example.com**

**二 搭建客户端,虚拟机Server0**

**1.安装软件sssd,与LDAP服务器沟通**

**[root@server0 /]# yum -y install sssd**

**2.安装图形软件配置sssd软件**

**[root@server0 /]# yum -y install authconfig-gtk**

**3.运行图形软件进行配置**

**[root@server0 /]# authconfig-gtk**

**选择LDAP**

**dc=example,dc=com #指定服务端域名**

**classroom.example.com #指定服务端主机名**

**勾选TLS加密**

**使用证书加密: http://classroom.example.com/pub/example-ca.crt**

**选择LDAP密码**

**4.重起服务**

**[root@server0 ~]# systemctl restart sssd**

**[root@server0 ~]# systemctl enable sssd**

**5.验证:**

**[root@server0 ~]# grep ldapuser0 /etc/passwd**

**[root@server0 ~]# id ldapuser0 #可以看到用户基本信息**

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

**家目录漫游**

什么是NFS共享

• Network File System,网络文件系统

– 由NFS服务器将指定的文件夹共享给客户机

– 客户机将此共享目录 mount 到本地目录,访问此共享

资源就像访问本地目录一样方便

– 类似于 EXT4、XFS等类型,只不过资源在网上

NFS共享服务器:classroom.example.com

客户端:虚拟机server0访问共享即可

**1.查看服务端共享文件夹**

**[root@server0 ~]# showmount -e classroom.example.com**

**Export list for classroom.example.com:**

**/home/guests 172.25.0.0/255.255.0.0**

**2.访问共享文件夹**

**# umount /nsd**

**# mkdir /home/guests**

**# mount classroom.example.com:/home/guests/ /home/guests**

**# ls /home/guests/**

**# su - ldapuser0**

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

**附加权限**

**Set UID**

• 附加在属主的 x 位上

– 属主的权限标识会变为 s

– 适用于可执行文件,Set UID可以让使用者具有文件属主的身份及部分权限

– 传递所有者身份

[root@server0 /]# /usr/bin/mkdir /opt/haha

[root@server0 /]# ls /opt/

[root@server0 /]# cp /usr/bin/mkdir /usr/bin/hahadir

[root@server0 /]# /usr/bin/hahadir /opt/xixi

[root@server0 /]# ls /opt/

[root@server0 /]# chmod u+s /usr/bin/hahadir

[root@server0 /]# ls -l /usr/bin/hahadir

[root@server0 /]# ls -l /usr/bin/mkdir

[root@server0 /]# su - zhangsan

[zhangsan@server0 ~]$ /usr/bin/mkdir test01

[zhangsan@server0 ~]$ ls -l

[zhangsan@server0 ~]$ /usr/bin/hahadir test02

[zhangsan@server0 ~]$ ls -l

[zhangsan@server0 ~]$ exit

logout

[root@server0 /]#

**Sticky Bit**

**• 附加在其他人的 x 位上**

**– 其他人的权限标识会变为 t**

**– 适用于开放 w 权限的目录**

**– 可以阻止用户滥用 w 写入权限(禁止操作别人的文档**)

[root@server0 /]# mkdir /public

[root@server0 /]# chmod ugo=rwx /public

[root@server0 /]# ls -ld public/

[root@server0 /]# chmod o+t /public/

[root@server0 /]# ls -ld /public/

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

**查找文件**

**• 根据预设的条件递归查找对应的文件**

**– find [目录] [条件1] [-a|-o] [条件2] ...**

**– 常用条件表示:**

**-type 类型(f文件、d目录、l快捷方式)**

**-name "文档名称"**

**-size +|-文件大小(k、M、G)**

**-user 用户名**

[root@server0 /]# find /boot/ -type l #查找是快捷方式

[root@server0 /]# ls /boot/grub/menu.lst

[root@server0 /]# ls -l /boot/grub/menu.lst

[root@server0 /]# find /boot/ -type d #查找是目录

[root@server0 /]# find /boot/ -type f #查找是文件

**请查找/etc目录下以".conf"结尾(包含子目录)**

[root@server0 ~]# find /etc/ -name "\*.conf"

[root@server0 ~]# find /etc/ -name "passwd"

[root@server0 ~]# find /etc/ -name "\*tab"

[root@server0 ~]# find /etc/ -name "shdaow"

[root@server0 ~]# find /etc/ -name "group"

[root@server0 ~]# find /etc/ -name "fstab"

[root@server0 ~]# find /etc/ -name "\*tab\*"

[root@server0 ~]# find /etc/ -name "resolv.conf"

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

[root@server0 /]# mkdir /root/nsd01

[root@server0 /]# mkdir /root/nsd02

[root@server0 /]# touch /root/nsd1806.txt

# find /root/ -name "nsd\*" -type d

# find /root/ -name "nsd\*" -type f

**-o 两个条件满足其中同一个即可**

**# find /root/ -name "nsd\*" -o -type d**

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

**-size +|-文件大小(k、M、G)**

**-user 用户名 #按照所有者进行查找**

[root@server0 /]# find /boot/ -size +20M

[root@server0 /]# find /boot/ -size +300k

[root@server0 /]# find /boot/ -size -300k

[root@server0 /]# find /home -user zhangsan

[root@server0 /]# find /home -user lisi

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

**查找文件**

**• 使用find命令的 -exec 操作**

**– find .. .. -exec 处理命令 {} \;**

**– 优势:以 {} 代替每一个结果,逐个处理,遇 \; 结束**

# find /boot/ -name "vm\*"

# find /boot/ -name "vm\*" -exec cp -r {} /opt \;

# ls /opt/

# find /boot/ -size +10M

# find /boot/ -size +10M -exec cp -r {} /opt \;

# ls /opt/

# find /etc/ -name "\*tab"

# find /etc/ -name "\*tab" -exec cp -r {} /opt \;

# ls /opt/

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

**• 使用find命令完成以下任务**

**– 找出所有用户 student 拥有的文件**

**– 把它们拷贝到 /root/findfiles/ 文件夹中**

[root@server0 /]# mkdir /root/findfiles

[root@server0 /]# ls /root/

# find / -user student -type f

# find / -user student -type f -exec cp {} /root/

/findfiles/ \;

[root@server0 /]# ls -A /root/findfiles/

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

**• 根据名称查找,忽略大小写**

**– -iname**

[root@server0 /]# find /etc/ -name "PASSWD"

[root@server0 /]# find /etc/ -iname "PASSWD"

• **根据所属组**

**-group**

[root@server0 /]# find /home/ -group student

**• 限制目录查找的深度(最大层数)**

**– -maxdepth**

[root@server0 /]# find /etc/ -maxdepth 1 -name "\*.conf"

[root@server0 /]# find /etc/ -maxdepth 2 -name "\*.conf"

[root@server0 /]# find /etc/ -maxdepth 3 -name "\*.conf"

• **根据文件修改时间,所有的时间都是过去时间**

**-mtime +|-**

**-mtime +10 #10天之前**

**-mtime -10 #最近10天之内**

**-mtime +90 #90天之前**

[root@server0 /]# find /var/log/ -mtime +100

[root@server0 /]# find /var/log/ -mtime -2

[root@server0 /]# find /var/log/ -mtime -10

[root@server0 /]# find /var/log/ -mtime +90

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

**查找文本内容**

**• 根据字符串模式提取文本行**

**– grep [选项] '匹配模式' 文本文件**

**– 命令行 | grep [选项] '匹配模式'**

**• 常用命令选项**

**– -v,取反匹配**

**– -i,忽略大小写**

[root@server0 /]# grep root /etc/passwd

[root@server0 /]# grep man /etc/man\_db.conf

[root@server0 /]# grep -i man /etc/man\_db.conf

[root@server0 /]# grep -v root /etc/passwd

**– ^root 以字符串root开头**

**– root$ 以字符串root结尾**

**– ^$ 匹配空行**

[root@server0 /]# grep root /etc/passwd

[root@server0 /]# grep ^root /etc/passwd

[root@server0 /]# grep root$ /etc/passwd

[root@server0 /]# grep bash$ /etc/passwd

显示文件有效配置(去掉注释,去掉空行)

# grep -v ^# /etc/login.defs | grep -v ^$

# grep -v ^# /etc/login.defs | grep -v ^$ > /opt/b.txt

# vim /opt/b.txt

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

Yum服务:为客户端安装软件包,自动解决依赖关系

服务端: 1.有软件包 2.仓库数据文件 3.共享服务(Web或FTP)

客户端: 书写一个配置文件 /etc/yum.repos.d/\*.repo

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

Question1:

在crontab文件中执行任务

idconfig >> /opt/1.txt

任务时没有定时输出内容到文档中为什么？

\* \* \* \* \* /usr/sbin/ifconfig >> /opt/1.txt

建立计划任务中所有命令都要是绝对路径

[root@server0 /]# which ifconfig #搜索命令所对应的执行程序

**二、Engineer**

**2018-07-09（第一天）**

教学环境介绍

[root@room9pc01 ~]# rht-vmctl reset classroom

classroom [OK]

域 classroom 已开始

[root@room9pc01 ~]# rht-vmctl reset server

server [OK]

域 server 已开始

[root@room9pc01 ~]# gos

Last login: Fri Jan 12 18:54:14 2018 from 172.25.0.250

/usr/bin/xauth: file /root/.Xauthority does not exist

[root@server0 ~]#

[root@server0 ~]# yum clean all #清空Yum缓存

[root@server0 ~]# yum repolist

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

服务端: 1.有软件包 2.仓库数据文件 3.共享服务(Web或FTP)

客户端: 书写一个配置文件 /etc/yum.repos.d/\*.repo

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

扇区默认的大小512字节

一块硬盘的“艺术”之旅

• 识别硬盘 => 分区规划 => 格式化 => 挂载使用

毛坯楼层 => 打隔断 => 装修 =>入驻

一 识别硬盘

[root@server0 ~]# **lsblk #列出当前系统所有识别的硬盘**

二 分区规划

**• MBR/msdos 分区模式**

**– 主分区 扩展分区 逻辑分区**

**– 1~4个主分区 或者 3个主分区+1个扩展分区(n个逻辑分区)**

**– 最大支持容量为 2.2TB 的磁盘**

**– 扩展分区不能格式化**

• **GPT 分区模式**

**– 最大支持容量为 18EB 的空间**

**1EB=1000PB**

**1PB=1000TB**

**– 划分128主分区**

厂商: 1TB=1000GB

计算机: 1TiB=1024GiB

**• 查看分区表**

**– fdisk -l /dev/sda**

• 修改硬盘的分区表

– fdisk 硬盘设备

[root@server0 ~]# fdisk /dev/vdb

**n 创建新的分区----->回车----->回车---->回车----->在last结束时 +1G**

**p 查看分区表**

**n 创建新的分区----->回车----->回车---->回车----->在last结束时 +2G**

**d 删除分区**

**w 保存并退出**

[root@server0 ~]# ls /dev/vdb[1-2]

[root@server0 ~]# lsblk

三  **格式化**

**[root@server0 ~]# mkfs.ext4 /dev/vdb1**

**[root@server0 ~]# mkfs.xfs /dev/vdb2**

**[root@server0 ~]# blkid /dev/vdb1 #可以看到UUID与文件系统**

**[root@server0 ~]# blkid /dev/vdb2 #可以看到UUID与文件系统**

UUID:设备的唯一标识

四 mount使用

[root@server0 ~]# mkdir /mypart1

[root@server0 ~]# mount /dev/vdb1 /mypart1

[root@server0 ~]# df -h #查看当前系统 已经挂载 的所有设备信息

[root@server0 ~]# mount /dev/vdb2 /mypart2

mount: 挂载点 /mypart2 不存在

[root@server0 ~]# mkdir /mypart2

[root@server0 ~]# mount /dev/vdb2 /mypart2

[root@server0 ~]# df -h

**五 开机自动挂载文件 /etc/fstab(file system table)**

补充:vim技巧: 安装 o 另起新的一行进入 输入模式(编辑模式)

设备路径 挂载点 类型 参数(支持的功能) 备份标记 检测顺序

/dev/vdb1 /mypart1 ext4 defaults 0 0

[root@server0 ~]# vim /etc/fstab

.........

/dev/vdb1 /mypart1 ext4 defaults 0 0

/dev/vdb2 /mypart2 xfs defaults 0 0

[root@server0 ~]# cat /etc/fstab

验证:

[root@server0 ~]# umount /mypart1

[root@server0 ~]# umount /mypart2

[root@server0 ~]# mount -a

1.需要将检测的设备,卸载

2.将当前 系统没有挂载,但/etc/fstab文件中书写了,设备进行挂载

[root@server0 ~]# df -h

[root@server0 ~]# mount -a

[root@server0 ~]# df -h

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

**总结:**

**1.列出识别硬盘 lsblk**

**2.划分新分区 fdisk**

**3.刷新分区表 partprobe**

**4.格式化 mkfs.ext4 mkfs.xfs ----> blkid 查看文件系统**

**5.挂载使用 mount ------> df -h 进行查看**

**6.开机自动挂载 /etc/fstab**

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

综合分区:

[root@server0 ~]# fdisk /dev/vdb

3个主分区 分别为 1G 2G 2G

划分扩展分区 占用所有剩余空间

在划分2个逻辑分区 2G 2G

**p 查看分区表**

**n 创建主分区----->回车----->回车---->回车----->在last结束时 +2G**

**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+2G**

**n 创建逻辑分区----->起始回车------>结束+2G**

**p 查看分区表**

**w 保存并退出**

[root@server0 ~]# lsblk

**[root@server0 ~]# partprobe #刷新磁盘分区**

[root@server0 ~]# lsblk

请描述该设备/dev/sda5标识的含义:

SCSI类型的设备,第一个硬盘的第五个分区

SCSI类型的设备,第一个硬盘的第一个逻辑分区

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

综合分区:

关机虚拟机server0:添加一块80G硬盘

[root@room9pc01 ~]# gos

[root@server0 ~]# lsblk #查看当前系统中识别的硬盘

[root@server0 ~]# fdisk /dev/vdc

划分6个10G可以使用的分区

p 查看分区表

n 创建主分区----->回车----->回车---->回车----->在last结束时 +10G

连续创建3个10G主分区

......

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+10G

n 创建逻辑分区----->起始回车------>结束+10G

n 创建逻辑分区----->起始回车------>结束+10G

p 查看分区表

w 保存并退出

[root@server0 ~]# ls /dev/vdc[1-7]

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

**LVM逻辑卷管理**

**逻辑卷作用:**

**1.整合分散的空间 2.可以动态扩大空间**

逻辑卷创建:

零散空闲存储 ----> 整合的虚拟磁盘---->虚拟的分区-->格式化--->挂载使用

物理卷(PV)------->卷组(VG)----->逻辑卷(LV)

将众多的物理卷(PV),组成卷组(VG),再从卷组中分出逻辑卷(LV)

面粉------>大面团-------->小面团------->蒸----->吃

砖-------->大房子--------->打隔断------>装修--->入住

物理卷lv

Physical Volume

卷组

Volume Group

逻辑卷

Logical Volume

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

**一 创建逻辑卷**

**1.创建卷组**

**格式：vgcreate 卷组名 设备路径**

**[root@server0 ~]# vgcreate systemvg /dev/vdc[1-2]**

[root@server0 ~]# vgs #显示卷组基本信息

[root@server0 ~]# pvs #显示物理卷基本信息

**2.创建逻辑卷**

**格式：lvcreate -n 逻辑卷名字 -L 大小 卷组名**

**kfs**

[root@server0 ~]# lvcreate -n mylv -L 16G systemvg

[root@server0 ~]# lvs

[root@server0 ~]# vgs

3.逻辑卷的使用

[root@server0 ~]# mkfs.ext4 /dev/systemvg/mylv

[root@server0 ~]# vim /etc/fstab

/dev/systemvg/mylv /lv ext4 defaults 0 0

**[root@server0 ~]# mount -a #进行自动挂载**

[root@server0 ~]# df -h

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

扩展逻辑卷

一 卷组有足够的剩余空间

**1.直接扩展逻辑卷的空间**

**[root@server0 ~]# lvextend -L 18G /dev/systemvg/mylv**

[root@server0 ~]# lvs #显示逻辑卷信息

[root@server0 ~]# vgs #显示卷组信息

**2.扩展逻辑卷的文件系统**

**刷新ext4文件系统：resize2fs**

**刷新xfs文件系统：xfs\_growfs**

**[root@server0 ~]# df -h**

**[root@server0 ~]# resize2fs /dev/systemvg/m**ylv

[root@server0 ~]# df -h

**二 卷组没有足够的剩余空间**

**1.扩展卷组空间**

**[root@server0 ~]# vgextend systemvg /dev/vdc3**

[root@server0 ~]# vgs

**2.直接扩展逻辑卷的空间**

**[root@server0 ~]# lvextend -L 25G /dev/systemvg/mylv**

[root@server0 ~]# lvs #显示逻辑卷信息

[root@server0 ~]# vgs #显示卷组信息

**3.扩展逻辑卷的文件系统**

**刷新ext4文件系统：resize2fs**

**刷新xfs文件系统：xfs\_growfs**

[root@server0 ~]# df -h

**[root@server0 ~]# resize2fs /dev/systemvg/mylv**

[root@server0 ~]# df -h

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

逻辑卷可以做缩减（了解）

ext4文件系统支持缩减

xfs文件系统不支持缩减

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

卷组划分空间的单位PE， 默认每一个PE大小4M

[root@server0 ~]# vgdisplay systemvg

• 创建卷组的时候设置PE大小

– vgcreate -s PE大小 卷组名 空闲分区...

– vgchange -s PE大小 卷组名 空闲分区...

**• 创建逻辑卷的时候指定PE个数**

**– lvcreate -l PE个数 -n 逻辑卷名 卷组名**

– 新的逻辑卷命名为 database,大小为50个物理扩展单

元(Physical Extent),属于 systemvg 卷组

– 在 datastore 卷组中的所有逻辑卷,其物理扩展单元

(Physical Extent)的大小为 1MiB

– 使用 EXT3 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

**1.显示卷组详细信息**

**[root@server0 ~]# vgdisplay systemvg #查看PE大小**

**2.修改卷组PE大小**

**[root@server0 ~]# vgchange -s 1M systemvg**

[root@server0 ~]# vgdisplay systemvg #查看PE大小

**3.以PE的个数，创建逻辑卷**

**[root@server0 ~]# lvcreate -l 50 -n database systemvg**

[root@server0 ~]# lvs

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

**逻辑卷删除**

**先删除逻辑卷，再删除卷组，最后删除物理卷**

[root@server0 ~]# lvremove /dev/systemvg/mylv

Logical volume systemvg/mylv contains a filesystem in use.

[root@server0 ~]# umount /lv

[root@server0 ~]# lvremove /dev/systemvg/mylv

Do you really want to remove active logical volume mylv? [y/n]: y

Logical volume "mylv" successfully removed

[root@server0 ~]# lvs

[root@server0 ~]# vgremove systemvg

Volume group "systemvg" successfully removed

[root@server0 ~]# vgs

[root@server0 ~]# pvs

[root@server0 ~]# pvremove /dev/vdc[1-3]

**2018-07-09（第一天作业）**

案例练习,准备:

rht-vmctl reset classroom

rht-vmctl reset server

rht-vmctl reset desktop

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

案例1:为虚拟机 server 配置以下永久的静态地址参数

– 主机名:server0.example.com

– IP地址:172.25.0.11

– 子网掩码:255.255.255.0

– 默认网关:172.25.0.254

– DNS服务器:172.25.254.254

[root@server0 ~]# echo server0.example.com > /etc/hostname

[root@server0 ~]# nmcli connection modify 'System eth0' ipv4.method manual ipv4.addresses '172.25.0.11/24 172.25.0.254' ipv4.dns 172.25.254.254 connection.autoconnect yes

案例2：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

[root@server0 ~]# mkdir /example

[root@server0 ~]# touch /example/nsd.txt

1）将“I love Study”写入到文件nsd.txt

[root@server0 ~]# echo I love Study > /example/nsd.txt

[root@server0 ~]# cat /example/nsd.txt

I love Study

2）将nsd.txt重命名为mylove.txt

[root@server0 ~]# mv /example/nsd.txt /example/mylove.txt

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

[root@server0 ~]# cp -r /etc/passwd /boot /etc/group /example

4）将ifconfig命令的前两行内容，追加写入mylove.txt

[root@server0 ~]# echo ifconfig >> /example/mylove.txt

5）将主机名永久配置文件，拷贝到/example/目录下

[root@server0 ~]# cp /etc/hostname /example

6）将DNS永久配置文件，拷贝到/example/目录下

[root@server0 ~]# cp /etc/resolv.conf /example

7）将开机自动挂载配置文件，拷贝到/example/目录下

[root@server0 ~]# cp /etc/fstab /example

案例3:在server上操作,（MBR分区模式）规划分区

添加一块80G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@server0 ~]# fdisk /dev/vdc

欢迎使用 fdisk (util-linux 2.23.2)。

更改将停留在内存中，直到您决定将更改写入磁盘。

使用写入命令前请三思。

Device does not contain a recognized partition table

使用磁盘标识符 0x68515094 创建新的 DOS 磁盘标签。

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):

Using default response p

分区号 (1-4，默认 1)：

起始 扇区 (2048-167772159，默认为 2048)：

将使用默认值 2048

Last 扇区, +扇区 or +size{K,M,G} (2048-167772159，默认为 167772159)：+10G

分区 1 已设置为 Linux 类型，大小设为 10 GiB

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):

Using default response p

分区号 (2-4，默认 2)：

起始 扇区 (20973568-167772159，默认为 20973568)：

将使用默认值 20973568

Last 扇区, +扇区 or +size{K,M,G} (20973568-167772159，默认为 167772159)：+10G

分区 2 已设置为 Linux 类型，大小设为 10 GiB

命令(输入 m 获取帮助)：N

Partition type:

p primary (2 primary, 0 extended, 2 free)

e extended

Select (default p):

Using default response p

分区号 (3,4，默认 3)：

起始 扇区 (41945088-167772159，默认为 41945088)：

将使用默认值 41945088

Last 扇区, +扇区 or +size{K,M,G} (41945088-167772159，默认为 167772159)：+12G

分区 3 已设置为 Linux 类型，大小设为 12 GiB

命令(输入 m 获取帮助)：n

Partition type:

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

e extended

Select (default e): e

已选择分区 4

起始 扇区 (67110912-167772159，默认为 67110912)：

将使用默认值 67110912

Last 扇区, +扇区 or +size{K,M,G} (67110912-167772159，默认为 167772159)：

将使用默认值 167772159

分区 4 已设置为 Extended 类型，大小设为 48 GiB

命令(输入 m 获取帮助)：n

All primary partitions are in use

添加逻辑分区 5

起始 扇区 (67112960-167772159，默认为 67112960)：

将使用默认值 67112960

Last 扇区, +扇区 or +size{K,M,G} (67112960-167772159，默认为 167772159)：+20G

分区 5 已设置为 Linux 类型，大小设为 20 GiB

命令(输入 m 获取帮助)：p

磁盘 /dev/vdc：85.9 GB, 85899345920 字节，167772160 个扇区

Units = 扇区 of 1 \* 512 = 512 bytes

扇区大小(逻辑/物理)：512 字节 / 512 字节

I/O 大小(最小/最佳)：512 字节 / 512 字节

磁盘标签类型：dos

磁盘标识符：0x68515094

设备 Boot Start End Blocks Id System

/dev/vdc1 2048 20973567 10485760 83 Linux

/dev/vdc2 20973568 41945087 10485760 83 Linux

/dev/vdc3 41945088 67110911 12582912 83 Linux

/dev/vdc4 67110912 167772159 50330624 5 Extended

/dev/vdc5 67112960 109055999 20971520 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。

案例4:在server上操作,分区使用

1、案例3中新添加80G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

[root@server0 ~]# mkdir /mnt/xfs

[root@server0 ~]# mkfs.xfs /dev/vdc5

meta-data=/dev/vdc5 isize=256 agcount=4, agsize=1310720 blks

= sectsz=512 attr=2, projid32bit=1

= crc=0

data = bsize=4096 blocks=5242880, imaxpct=25

= sunit=0 swidth=0 blks

naming =version 2 bsize=4096 ascii-ci=0 ftype=0

log =internal log bsize=4096 blocks=2560, version=2

= sectsz=512 sunit=0 blks, lazy-count=1

realtime =none extsz=4096 blocks=0, rtextents=0

[root@server0 ~]# vim /etc/fstab

[root@server0 ~]# mount -a

[root@server0 ~]# df -h

文件系统 容量 已用 可用 已用% 挂载点

/dev/vda1 10G 3.1G 7.0G 31% /

devtmpfs 906M 0 906M 0% /dev

tmpfs 921M 80K 921M 1% /dev/shm

tmpfs 921M 17M 904M 2% /run

tmpfs 921M 0 921M 0% /sys/fs/cgroup

/dev/vdc5 20G 33M 20G 1% /mnt/xfs

2、案例3中新添加80G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@server0 ~]# mkdir /mnt/mypart

[root@server0 ~]# mkfs.ext4 /dev/vdc1

mke2fs 1.42.9 (28-Dec-2013)

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

655360 inodes, 2621440 blocks

131072 blocks (5.00%) reserved for the super user

第一个数据块=0

Maximum filesystem blocks=2151677952

80 block groups

32768 blocks per group, 32768 fragments per group

8192 inodes per group

Superblock backups stored on blocks:

32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: 完成

正在写入inode表: 完成

Creating journal (32768 blocks): 完成

Writing superblocks and filesystem accounting information: 完成

[root@server0 ~]# vim /etc/fstab

[root@server0 ~]# mount -a

[root@server0 ~]# df -h

文件系统 容量 已用 可用 已用% 挂载点

/dev/vda1 10G 3.1G 7.0G 31% /

devtmpfs 906M 0 906M 0% /dev

tmpfs 921M 80K 921M 1% /dev/shm

tmpfs 921M 17M 904M 2% /run

tmpfs 921M 0 921M 0% /sys/fs/cgroup

/dev/vdc5 20G 33M 20G 1% /mnt/xfs

/dev/vdc1 9.8G 37M 9.2G 1% /mnt/mypart

案例5:虚拟机 server0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@server0 ~]# mkdir /vo

[root@server0 ~]# vgcreate systemvg /dev/vdc2

Physical volume "/dev/vdc2" successfully created

Volume group "systemvg" successfully created

[root@server0 ~]# lvcreate -n vo -L 8G systemvg

Logical volume "vo" created

[root@server0 ~]# mkfs.xfs /dev/systemvg/vo

meta-data=/dev/systemvg/vo isize=256 agcount=4, agsize=524288 blks

= sectsz=512 attr=2, projid32bit=1

= crc=0

data = bsize=4096 blocks=2097152, imaxpct=25

= sunit=0 swidth=0 blks

naming =version 2 bsize=4096 ascii-ci=0 ftype=0

log =internal log bsize=4096 blocks=2560, version=2

= sectsz=512 sunit=0 blks, lazy-count=1

realtime =none extsz=4096 blocks=0, rtextents=0

[root@server0 ~]# mount -a

[root@server0 ~]# df -h

文件系统 容量 已用 可用 已用% 挂载点

/dev/vda1 10G 3.1G 7.0G 31% /

devtmpfs 906M 0 906M

tmpfs 921M 80K 921M 1% /dev/shm

tmpfs 921M 17M 904M 2% /run

tmpfs 921M 0 921M 0% /sys/fs/cgroup

/dev/vdc5 20G 33M 20G 1% /mnt/xfs

/dev/vdc1 9.8G 37M 9.2G 1% /mnt/mypart

/dev/mapper/systemvg-vo 8.0G 33M 8.0G 1% /vo

案例6:虚拟机 server0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@server0 ~]# vgcreate datastore /dev/vdc3

Physical volume "/dev/vdc3" successfully created

Volume group "datastore" successfully created

[root@server0 ~]# vgchange -s 1M datastore

Volume group "datastore" successfully changed

[root@server0 ~]# lvcreate -l 50 -n database datastore

Logical volume "database" created

[root@server0 ~]# mkdir /mnt/datastore

[root@server0 ~]# mkfs.ext4 /dev/datastore/database

mke2fs 1.42.9 (28-Dec-2013)

文件系统标签=

OS type: Linux

块大小=1024 (log=0)

分块大小=1024 (log=0)

Stride=0 blocks, Stripe width=0 blocks

12824 inodes, 51200 blocks

2560 blocks (5.00%) reserved for the super user

第一个数据块=1

Maximum filesystem blocks=33685504

7 block groups

8192 blocks per group, 8192 fragments per group

1832 inodes per group

Superblock backups stored on blocks:

8193, 24577, 40961

Allocating group tables: 完成

正在写入inode表: 完成

Creating journal (4096 blocks): 完成

Writing superblocks and filesystem accounting information: 完成

[root@server0 ~]# vim /etc/fstab

[root@server0 ~]# mount -a

[root@server0 ~]# df -h

文件系统 容量 已用 可用 已用% 挂载点

/dev/vda1 10G 3.1G 7.0G 31% /

devtmpfs 906M 0 906M 0% /dev

tmpfs 921M 80K 921M 1% /dev/shm

tmpfs 921M 17M 904M 2% /run

tmpfs 921M 0 921M 0% /sys/fs/cgroup

/dev/vdc5 20G 33M 20G 1% /mnt/xfs

/dev/vdc1 9.8G 37M 9.2G 1% /mnt/mypart

/dev/mapper/systemvg-vo 8.0G 33M 8.0G 1% /vo

/dev/mapper/datastore-database 45M 1.1M 40M 3% /mnt/datastore

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@server0 ~]# vgextend systemvg /dev/vdc3

[root@server0 ~]# lvextend -L 20G /dev/systemvg/vo

[root@server0 ~]# xfs\_growfs /dev/systemvg/vo

案例8: vim文本编辑器练习，命令模式下的基本操作（重点知识）

1）将/etc/passwd的前20行，写入到文件/opt/pass20.txt

[root@localhost ~]# head -20 /etc/passwd > /opt/pass20.txt

2）将/etc/shadow的前10行,追加写入到文件/opt/pass20.txt

[root@localhost ~]# head -10 /etc/passwd >> /opt/pass20.txt

3）利用/opt/pass20.txt，进项如下操作：

– 命令模式下移动光标：键盘上下左右键、Home键、End键

– 命令模式下行间跳转：到全文的第一行（1G或gg）、到全文的最后一行（G）、到全文的第10行（10G）

– 命令模式下复制、粘贴：

复制1行（yy）、复制3行（3yy）

粘贴到当前行之后（小写p）

– 命令模式下删除：

删除单个字符（x）

删除到行首（d^）、删除到行尾（d$）

删除1行（dd）、删除3行（3dd）

– 命令模式下查找关键词：

搜索（/word）切换结果（n、N）

– 补充：在命令模式下大

写的C，可以删除光标之后，并且进入输入模式

**2018-07-10（第二天）**

教学环境介绍

[root@room9pc01 ~]# rht-vmctl reset classroom

classroom [OK]

域 classroom 已开始

[root@room9pc01 ~]# rht-vmctl reset server

server [OK]

域 server 已开始

[root@room9pc01 ~]# gos

Last login: Fri Jan 12 18:54:14 2018 from 172.25.0.250

/usr/bin/xauth: file /root/.Xauthority does not exist

[root@server0 ~]#

[root@server0 ~]# yum clean all #清空Yum缓存

[root@server0 ~]# yum repolist

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

Shell脚本基础

脚本/程序：可以执行文本文件，可以带来某种功能

规范Shell脚本的一般组成

• **#! 环境声明(以下代码由谁解释)**

• # 注释文本

• 可执行代码

案例1：书写hello.sh脚本,可以再屏幕输出hello world

[root@server0 ~]# vim /root/hello.sh

#!/bin/bash

echo hello world

cat /etc/redhat-release

hostname

uname -r

ifconfig | head -2

[root@server0 ~]# chmod +x /root/hello.sh

[root@server0 ~]# /root/hello.sh #绝对路径进行执行

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

管道传递

• 使用 | 管道操作

– 将前一条命令的标准输出交给后一条命令处理

重定向输出

> : 正确信息重定向,只收集前面命令的正确输出结果

2>: 错误信息重定向,只收集前面命令的错误输出结果

&>: 正确与错误信息重定向

[root@server0 ~]# echo 123 > /opt/1.txt

[root@server0 ~]# cat /opt/1.txt

[root@server0 ~]# cat /opt/1.txt /etc

[root@server0 ~]# cat /opt/1.txt /etc > /mnt/a.txt

[root@server0 ~]# cat /mnt/a.txt

[root@server0 ~]# cat /opt/1.txt /etc 2> /mnt/a.txt

[root@server0 ~]# cat /mnt/a.txt

[root@server0 ~]# cat /opt/1.txt /etc &> /mnt/a.txt

[root@server0 ~]# cat /mnt/a.txt

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

**单引号 ' ' ：取消所有特殊字符的含义**

**$[ ] :运算 + - \* / %(取余数运算)**

[root@server0 ~]# echo $[1+2]

[root@server0 ~]# echo $[8-6]

[root@server0 ~]# echo $[2\*3]

[root@server0 ~]# echo $[10/5]

[root@server0 ~]# echo $[10/3]

[root@server0 ~]# echo $[10%3]

[root@server0 ~]# echo $[10%2]

**余数一定小于除数**

**$( ) 与 反撇号 ` ` : 将命令的输出结果，作为参数 ， 参与命令的运行**

[root@server0 ~]# rm -rf /opt/\*

[root@server0 opt]# cd /opt

[root@server0 opt]# date +%F

[root@server0 opt]# mkdir stu-`date +%F`

[root@server0 opt]# ls

[root@server0 opt]# mkdir nsd-$(date +%F)

[root@server0 opt]# ls

[root@server0 opt]# mkdir $(hostname)-$(date +%F)

[root@server0 opt]# ls

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

案例2：创建用户的脚本

**黑洞设备: /dev/null**

[root@server0 /]# vim /root/user.sh

#!/bin/bash

useradd nsd10 &> /dev/null

echo 用户nsd10创建成功

echo 123456 | passwd --stdin nsd10 &> /dev/null

echo 用户nsd10密码设置成功

[root@server0 /]# chmod +x /root/user.sh

[root@server0 /]# /root/user.sh

为了增加脚本的灵活度，环境的多变性，方便重复使用一个值

变量：容器，储存的值在变化。以不变的名称，存放可以变化的值

[root@server0 /]# a=1000

[root@server0 /]# echo a

[root@server0 /]# echo $a #输出变量储存的值

[root@server0 /]# a=rhel7

[root@server0 /]# echo $a

[root@server0 /]# a=dc

[root@server0 /]# echo $a

[root@server0 /]# a=20

[root@server0 /]# echo $a

[root@server0 /]# vim /root/user.sh

#!/bin/bash

a=nsd15

useradd $a &> /dev/null

echo 用户$a创建成功

echo 123456 | passwd --stdin $a &> /dev/null

echo 用户$a密码设置成功

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

**定义变量**

**– 等号两边不要有空格**

**– 变量名只能由字母/数字/下划线组成,区分大小写**

**– 变量名不能以数字开头,不要使用关键字和特殊字符**

**– 若指定的变量名已存在,相当于为此变量重新赋值**

**查看/引用变量**

**• 基本格式**

**– 引用变量值:$变量名**

**– 查看变量值:echo $变量名、echo ${变量名}**

[root@server0 /]# a=rhel

[root@server0 /]# echo $a

rhel

[root@server0 /]# echo $a7

[root@server0 /]# echo ${a}7

rhel7

[root@server0 /]#

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

**环境变量 变量名一般都大写,用来设置用户/系统环境**

**位置变量 bash内置,存储执行脚本时提供的命令行参数**

**预定义变量 bash内置,可直接调用的特殊值,不能直接修改**

**自定义变量 用户自主设置、修改及使用**

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

环境变量,由系统定义并且赋值完成，用户直接调用

USER=当前登陆的用户名

RANDOM=永远存放一个随机数

[root@server0 /]# echo $USER

[root@server0 /]# su - nsd01

[nsd01@server0 ~]$ echo $USER

[nsd01@server0 ~]$ exit

[root@server0 /]# su - student

[student@server0 ~]$ echo $USER

[student@server0 ~]$ exit

[root@server0 /]# echo $RANDOM

**产生一个有范围的随机数字，取余数运算，余数一定小于除数**

[root@server0 /]# echo $[$RANDOM%10]

**#####################################################**

**位置变量**

**• 由系统定义并且赋值完成，用户直接调用**

**• 在执行脚本时提供的命令行参数**

**– 表示为 $n,n为序号**

– $1、$2、.. .. ${10}、${11}、.. ..

[root@server0 ~]# vim /root/user.sh

#!/bin/bash

useradd $1 &> /dev/null

echo 用户$1创建成功

echo 123456 | passwd --stdin $1 &> /dev/null

echo 用户$1密码设置成功

[root@server0 ~]# /root/user.sh stu01

[root@server0 ~]# /root/user.sh stu02

[root@server0 ~]# vim /root/2.sh

#!/bin/bash

cat -n $1 | head -$2

[root@server0 ~]# /root/2.sh /etc/passwd 3

[root@server0 ~]# /root/2.sh /etc/group 3

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

**预定义变量**

**• 由系统定义并且赋值完成，用户直接调用**

**• 用来保存脚本程序的执行信息**

**– 直接使用这些变量**

**– 不能直接为这些变量赋值**

**$# 统计已加载的位置变量的个数**

**$\* 所有位置变量的值**

**$? 程序退出后的状态值,0表示正常,其他值异常**

[root@server0 ~]# useradd stu07

[root@server0 ~]# echo $?

0

[root@server0 ~]# useradd stu07

useradd：用户“stu07”已存在

[root@server0 ~]# echo $?

9

[root@server0 ~]# vim /root/1.sh

#!/bin/bash

echo $1

echo $2

echo $#

echo $\*

[root@server0 ~]# /root/1.sh student classroom server

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

条件测试及选择

– [ 测试表达式 ] 每一部分到有空格

**常用的测试选项**

**• 检查文件状态**

**-e:文档存在为真**

**-d:文档存在,且必须为目录为真**

**-f:文档存在,且必须为文件为真**

[root@server0 ~]# [ -f /etc ]

[root@server0 ~]# echo $?

[root@server0 ~]# [ -d /etc ]

[root@server0 ~]# echo $?

[root@server0 ~]# [ -f /etc/passwd ]

[root@server0 ~]# echo $?

[root@server0 ~]# [ -f /etc/1.txt ]

[root@server0 ~]# echo $?

• **比较整数大小**

**-gt:大于**

**-ge:大于等于**

**-eq:等于**

**-ne:不等于**

**-lt:小于**

**-le:小于等于**

[root@server0 ~]# [ 1 -eq 1 ]

[root@server0 ~]# echo $?

[root@server0 ~]# [ 1 -ge 1 ]

[root@server0 ~]# echo $?

[root@server0 ~]# [ 1 -gt 1 ]

[root@server0 ~]# echo $?

[root@server0 ~]# [ 1 -le 2 ]

[root@server0 ~]# echo $?

• **字符串比对**

**== : 一致为真**

**!= : 不一致为真**

[root@server0 ~]# [ root == student ]

[root@server0 ~]# echo $?

[root@server0 ~]# [ root != student ]

[root@server0 ~]# echo $?

[root@server0 ~]# [ root == root ]

[root@server0 ~]# echo $?

[root@server0 ~]# echo $USER

[root@server0 ~]# [ root == $USER ]

[root@server0 ~]# echo $?

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

if双分支处理

if [条件测试];then

命令序列xx

else

命令序列yy

fi

if [ 1 -eq 2 ];then

echo hello

else

echo hi

fi

案例3：判断用户通过命令行给出参数，参数为IP地址

书写一个脚本判断，本机是否可以与用户输入的IP地址进行通信

如果能够通信，则输出 可以通信

如果不能够通信，则输出 不可以通信

[root@server0 ~]# vim /root/ip.sh

#!/bin/bash

ping -c 2 $1 &> /dev/null

if [ $? -eq 0 ];then

echo $1可以通信

else

echo $1不可以通信

fi

[root@server0 ~]# /root/ip.sh 192.168.1.1

[root@server0 ~]# /root/ip.sh 172.25.0.11

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

提升用户的体验

产生交互，与用户进行问答，将用户在键盘上输入的答案，作为参数

read:产生交互,将用户在键盘上输入的答案,存放到变量中

[root@server0 ~]# vim /root/ip.sh

#!/bin/bash

read -p '请输入您要测试的IP地址:' a

ping -c 2 $a &> /dev/null

if [ $? -eq 0 ];then

echo $a可以通信

else

echo $a不可以通信

fi

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

案例4：书写猜数的脚本

计算机随机产生一个 0至9之间的数字

用户输入一个一个 0至9之间的数字

如果 计算机随机产生的数字 与 用户输入的数字 相等，则输出 猜对了

如果 计算机随机产生的数字 与 用户输入的数字 不相等，则输出 猜错了

[root@server0 ~]# vim /root/num.sh

#!/bin/bash

num1=$[$RANDOM%10]

read -p '请输入一个0～9之间的数字：' num2

if [ $num1 -eq $num2 ];then

echo 恭喜您，猜对了

else

echo 猜错了，谢谢

echo 正确的数字为$num1

fi

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

**if多分支处理**

**if [条件测试1];then**

**命令序列xx**

**elif [条件测试2];then**

**命令序列yy**

**elif [条件测试3];then**

**命令序列aa**

**elif [条件测试4];then**

**命令序列bb**

**......**

**else**

**命令序列zz**

**fi**

案例5：书写成绩判断脚本

用户输入本次考试的成绩：

如果大于等于90，则输出 优秀

如果大于等于80，则输出 良好

如果大于等于70，则输出 一般

如果大于等于60，则输出 合格

如果以上条件均不满足，则输出 再牛的肖邦，也弹不出哥的悲伤

[root@server0 ~]# vim /root/if03.sh

#!/bin/bash

read -p '请输入您的成绩:' num

if [ $num -ge 90 ];then

echo 优秀

elif [ $num -ge 80 ];then

echo 良好

elif [ $num -ge 70 ];then

echo 一般

elif [ $num -ge 60 ];then

echo 合格

else

echo 再牛的肖邦，也弹不出哥的悲伤

fi

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

案例6：书写猜数的脚本

在案例4基础上，当用户没有猜对，进行判断输出 用户猜大了 或 猜小了

[root@server0 ~]# vim /root/num.sh

#!/bin/bash

num1=$[$RANDOM%10]

read -p '请输入一个0～9之间的数字：' num2

if [ $num1 -eq $num2 ];then

echo 恭喜您，猜对了

elif [ $num2 -gt $num1 ];then

echo 猜大了

echo 正确的数字为$num1

else

echo 猜小了

echo 正确的数字为$num1

fi

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

思考：实现用户可以连续猜三次机会

[root@server0 ~]# vim /root/num.sh

#!/bin/bash

num1=$[$RANDOM%10]

read -p '请输入一个0～9之间的数字：' num2

if [ $num1 -eq $num2 ];then

echo 恭喜您，猜对了

exit

elif [ $num2 -gt $num1 ];then

echo 猜大了

else

echo 猜小了

fi

read -p '请输入一个0～9之间的数字：' num2

if [ $num1 -eq $num2 ];then

echo 恭喜您，猜对了

exit

elif [ $num2 -gt $num1 ];then

echo 猜大了

else

echo 猜小了

fi

read -p '请输入一个0～9之间的数字：' num2

if [ $num1 -eq $num2 ];then

echo 恭喜您，猜对了

exit

elif [ $num2 -gt $num1 ];then

echo 猜大了

else

echo 猜小了

fi

**2018-07-11(第三天)：**

教学环境介绍

[root@room9pc01 ~]# rht-vmctl reset classroom

classroom [OK]

域 classroom 已开始

[root@room9pc01 ~]# rht-vmctl reset server

server [OK]

域 server 已开始

[root@room9pc01 ~]# gos

Last login: Fri Jan 12 18:54:14 2018 from 172.25.0.250

/usr/bin/xauth: file /root/.Xauthority does not exist

[root@server0 ~]#

[root@server0 ~]# yum clean all #清空Yum缓存

[root@server0 ~]# yum repolist

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

列表式循环,自动帮助用户，重复执行一个动作

for 变量名 in 值列表

do

命令序列

done

for a （献血车 ） in zhangsan lisi wangwu （队伍）

do

useradd $a （重复执行的操作）

done

[root@server0 ~]# vim /root/for01.sh

#!/bin/bash

for a in zhangsan lisi wangwu tom natasha

do

useradd $a

echo $a创建成功

done

[root@server0 ~]#

循环方面：

1. 循环的列表值，参与循环体

2. 循环的列表值，不参与循环体

造数工具： {起点..结束}

[root@server0 /]# vim /root/for01.sh

#!/bin/bash

num1=$[$RANDOM%10]

for a in {1..3}

do

read -p '请输入0～9之间的一个数字:' num2

if [ $num1 -eq $num2 ];then

echo 恭喜您，猜对啦

exit

elif [ $num2 -gt $num1 ];then

echo 您猜大了

else

echo 您猜小了

fi

done

[root@server0 /]

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

系统安全保护

SELinux安全机制

• 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

**• SELinux的运行模式**

**– enforcing(强制)、permissive(宽松)**

**– disabled(彻底禁用)**

**所有的模式进入disabled模式，都要经过重起系统**

• **切换运行模式**

**– 查看当前SELinux模式：getenforce**

**– 临时切换:setenforce 1|0**

**每次开机默认的模式：**

**– 固定配置:/etc/selinux/config 文件**

虚拟机Server0

1.当前修改

[root@server0 /]# getenforce #查看SELinux当前的状态

Enforcing

[root@server0 /]# setenforce 0 #修改SELinux当前的状态

[root@server0 /]# getenforce

Permissive

**2.修改配置文件/etc/selinux/config**

**SELINUX=permissive**

虚拟机desktop0，做同样的修改

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

配置用户环境

alias别名设置

• 查看已设置的别名

– alias [别名名称]

• 定义新的别名

– alias 别名名称= '实际执行的命令行'

• 取消已设置的别名

– unalias [别名名称]

• 影响指定用户的 bash 解释环境

– ~/.bashrc,每次开启 bash 终端时生效

• 影响所有用户的 bash 解释环境

– /etc/bashrc,每次开启 bash 终端时生效

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

**[root@server0 /]# vim /root/.bashrc #只针对与root用户**

**alias hi='echo hi'**

**[root@server0 /]# vim /etc/bashrc #针对所有用户**

**alias hello='echo hello'**

[root@server0 /]# vim /home/student/.bashrc

alias haxi='echo hahaxixi'

[root@server0 /]# su - student

[student@server0 ~]$ hi #执行失败

[student@server0 ~]$ hello #执行成功

[student@server0 ~]$ haxi #执行成功

[student@server0 ~]$ exit

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

防火墙策略管理

**一、搭建基本Web服务**

Web服务器：虚拟机Server0

**1.安装软件包**  Apache(httpd)

[root@server0 /]# yum -y install httpd

**2.启动httpd服务**

[root@server0 /]# systemctl restart httpd

[root@server0 /]# systemctl enable httpd

ln -s '/usr/lib/systemd/system/httpd.service' '/etc/systemd/system/multi-user.target.wants/httpd.service'

**3.本机测试**

**[root@server0 /]# firefox 172.25.0.11**

**4.书写网页文件**

**默认存放网页路径：/var/www/html**

**默认网页文件名字：index.html**

[root@server0 ~]# vim /var/www/html/index.html

<marquee><font color=red><h1>NSD1806 阳光明媚</marquee>

滚动 字体颜色为红色 最大字体

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

**搭建FTP服务**

虚拟机Server0：

**1.安装vsftpd软件**

[root@server0 ~]# yum info vsftpd #查看软件包的基本信息

[root@server0 ~]# yum -y install vsftpd

**2.重起vsftpd服务**

[root@server0 ~]# systemctl restart vsftpd

[root@server0 ~]# systemctl enable vsftpd #设置开机自启动

ln -s '/usr/lib/systemd/system/vsftpd.service' '/etc/systemd/system/multi-user.target.wants/vsftpd.service'

**3.本机测试**

默认FTP服务共享路径：/var/ftp

**[root@server0 ~]# firefox ftp://172.25.0.11**

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

RHEL7防火墙

防火墙作用： 隔离， 默认允许出站 ，过滤入站

硬件防火墙

软件防火墙

RHEL7防火墙 :firewalld

• **管理工具:firewall-cmd、firewall-config(图形工具)**

预设安全区域

• 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sshd、ping、DHCP服务**

**– trusted:允许任何访问**

**– block:阻塞任何来访请求，明确拒绝**

**– drop:丢弃任何来访的数据包，不给出回应，节省资源**

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

防火墙的判定规则：匹配及停止

1.查看数据包中源IP地址，检查所有区域中，那个区域由改源IP地址的规则

则进入该区域

2.进入默认区域

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

防火墙默认区域的修改

虚拟机Server0

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

虚拟机Desktop0

# ping 172.25.0.11 #可以通信

虚拟机Server0

**# firewall-cmd --set-default-zone=block #修改默认区域**

# firewall-cmd --get-default-zone

虚拟机Desktop0

# ping 172.25.0.11 #不可以通信，有回应

虚拟机Server0

# firewall-cmd --set-default-zone=drop #修改默认区域

# firewall-cmd --get-default-zone

虚拟机Desktop0

# ping 172.25.0.11 #不可以通信，没有回应

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

默认区域 服务添加

虚拟机Server0

# firewall-cmd --set-default-zone=public #修改默认区域

**# firewall-cmd --zone=public --list-all #查看public区域规则**

虚拟机Desktop0

# firefox 172.25.0.11 #访问失败

# firefox ftp://172.25.0.11 #访问失败

虚拟机Server0

**# firewall-cmd --zone=public --add-service=http #添加http协议**

**# firewall-cmd --zone=public --list-all #查看public区域规则**

虚拟机Desktop0

# firefox 172.25.0.11 #访问成功

# firefox ftp://172.25.0.11 #访问失败

虚拟机Server0

**# firewall-cmd --zone=public --add-service=ftp #添加ftp协议**

**# firewall-cmd --zone=public --list-all #查看public区域规则**

虚拟机Desktop0

# firefox 172.25.0.11 #访问成功

# firefox ftp://172.25.0.11 #访问成功

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

**互联网常见的应用服务协议**

**http:超文本传输协议 端口号：80**

**FTP：文件传输协议 端口号：21**

**https:安全超文本传输协议 端口号：443**

**DNS:域名解析协议 端口号：53**

**tftp:简单的文件传输协议 端口号：69**

**telnet:远程管理协议 端口号：23**

**smtp:发邮件协议 端口号：25**

**pop3:收邮件协议 端口号：110**

**snmp:互联管理协议 端口号：161**

端口：编号标识协议（程序、服务）

数据包： 源IP地址 目标IP地址 数据 访问的端口号

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

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

**• 永久配置规则(permanent)**

**# firewall-cmd --reload #重新加载防火墙所有配置**

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

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

# firewall-cmd --reload

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

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

# firewall-cmd --reload

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

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

实现本机的端口映射

• 本地应用的端口重定向(端口1 --> 端口2)

– 从客户机访问 端口1 的请求,自动映射到本机 端口2

– 比如,访问以下两个地址可以看到相同的页面:

http://172.25.0.11:5423

http://172.25.0.11:80

虚拟机Server0

# firewall-cmd --permanent --zone=public

--add-forward-port=port=5423:proto=tcp:toport=80

# firewall-cmd --reload

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

虚拟机Desktop0

[root@desktop0 ~]# firefox 172.25.0.11:5423

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

当客户端访问本机，协议为tcp端口号为5423，将其端口修改为访问80端口

**2018-07-11（当天作业）**

案例1:为虚拟机 server 配置以下静态地址参数

– 主机名:server0.example.com

– IP地址:172.25.0.11

– 子网掩码:255.255.255.0

– 默认网关:172.25.0.254

– DNS服务器:172.25.254.254

[root@server0 ~]# echo server0.example.com > /etc/hostname

[root@server0 ~]# nmcli connection modify 'System eth0' ipv4.method manual ipv4.addresses '172.25.0.11/24 172.25.0.254' ipv4.dns 172.25.254.254 connection.autoconnect yes

案例2：设置SELinux保护

为虚拟机 server0、desktop0 配置SELinux

1）确保 SELinux 处于强制启用模式（permissive）

2）在每次重新开机后，此设置必须仍然有效

[root@server0 ~]# setenforce 0

[root@server0 ~]# getenforce

Permissive

[root@server0 ~]# vim /etc/selinux/config

[root@server0 ~]# head -7 /etc/selinux/config

# This file controls the state of SELinux on the system.

# SELINUX= can take one of these three values:

# enforcing - SELinux security policy is enforced.

# permissive - SELinux prints warnings instead of enforcing.

# disabled - No SELinux policy is loaded.

SELINUX=permissive

[root@desktop0 ~]# setenforce 0

[root@desktop0 ~]# getenforce

Permissive

[root@desktop0 ~]# vim /etc/selinux/config

[root@desktop0 ~]# head -7 /etc/selinux/config

# This file controls the state of SELinux on the system.

# SELINUX= can take one of these three values:

# enforcing - SELinux security policy is enforced.

# permissive - SELinux prints warnings instead of enforcing.

# disabled - No SELinux policy is loaded.

SELINUX=permissive

案例3：自定义用户环境

为系统 server0 创建自定义命令

1）自定义命令的名称为 qstat

2）此自定义命令将执行以下操作： /bin/ps -Ao pid,tt,user,fname,rsz

3）此自定义命令对系统中的所有用户都有

[root@server0 ~]# vim /etc/bashrc

[root@server0 ~]# head -2 /etc/bashrc

# /etc/bashrc

alias qstat='/bin/ps -Ao pid,tt,user,fname,rsz'

案例4：Shell脚本

为系统 server0 书写脚本/root/user.sh

运行脚本，可以判断用户输入的用户是否存在

如果存在，输出用户基本信息（id 用户名）

如果用户，不存在则创建用户，并输出用户创建成功

[root@server0 ~]# vim /root/uesr.sh

[root@server0 ~]# chmod +x /root/uesr.sh

[root@server0 ~]# head -10 /root/uesr.sh

#!/bin/bash

read -p '请输入用户名：' num

id $num &> /dev/null

if [ $? -eq 0 ];then

id $num

else

useradd $num &> /dev/null

echo 用户$num创建成功

fi

案例5：实现本机server0 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

[root@server0 ~]# yum -y install httpd

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# systemctl enable httpd

[root@server0 ~]# vim /var/www/html/index.html

小蝌蚪找妈妈

案例6：实现本机server0 的防火墙配置

1）修改虚拟机 server0防火墙配置，明确拒绝所有客户端访问

[root@server0 ~]# firewall-cmd --set-default-zone=block

success

[root@server0 ~]# firewall-cmd --get-default-zone

block

2）在虚拟机 desktop0上测试能否访问server0 的Web服务

[root@desktop0 ~]# firefox 172.25.0.11

不能访问

3）在虚拟机 desktop0上测试能否 ping通 虚拟机 server0

[root@desktop0 ~]# ping -c 4 172.25.0.11

PING 172.25.0.11 (172.25.0.11) 56(84) bytes of data.

From 172.25.0.11 icmp\_seq=1 Destination Host Prohibited

From 172.25.0.11 icmp\_seq=2 Destination Host Prohibited

From 172.25.0.11 icmp\_seq=3 Destination Host Prohibited

From 172.25.0.11 icmp\_seq=4 Destination Host Prohibited

不能拼通

案例7：实现本机server0 的防火墙配置

1）修改虚拟机 server0防火墙配置，将默认区域修改为public

[root@server0 ~]# firewall-cmd --set-default-zone=public

success

[root@server0 ~]# firewall-cmd --get-default-zone

public

2）在虚拟机 desktop0上测试能否访问server0 的Web服务

[root@desktop0 ~]# firefox 172.25.0.11

不能访问

3）在虚拟机 desktop0上测试能否 ping通 虚拟机 server0

[root@desktop0 ~]# ping -c 4 172.25.0.11

PING 172.25.0.11 (172.25.0.11) 56(84) bytes of data.

64 bytes from 172.25.0.11: icmp\_seq=1 ttl=64 time=0.477 ms

64 bytes from 172.25.0.11: icmp\_seq=2 ttl=64 time=0.491 ms

64 bytes from 172.25.0.11: icmp\_seq=3 ttl=64 time=0.575 ms

64 bytes from 172.25.0.11: icmp\_seq=4 ttl=64 time=0.506 ms

可以拼通

案例8：实现本机server0 的防火墙配置

1）修改虚拟机 server0防火墙配置，将默认区域修改为public

[root@server0 ~]# firewall-cmd --set-default-zone=public

success

[root@server0 ~]# firewall-cmd --get-default-zone

public

2）修改虚拟机 server0防火墙配置，在public区域中添加http服务

[root@server0 ~]# firewall-cmd --zone=public --add-service=http

success

[root@server0 ~]# firewall-cmd --zone=public --list-all

public (default, active)

interfaces: eth0

sources:

services: dhcpv6-client http ssh

ports:

masquerade: no

forward-ports:

icmp-blocks:

rich rules:

3）在虚拟机 desktop0上测试能否访问server0 的Web服务

[root@desktop0 ~]# firefox 172.25.0.11

可以访问

**2018-07-12（第四天）：**

教学环境介绍

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server

[root@room9pc01 ~]# rht-vmctl reset desktop

[root@room9pc01 ~]# gos

[root@server0 ~]#

[root@server0 ~]# yum clean all #清空Yum缓存

[root@server0 ~]# yum repolist

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

案例4：Shell脚本

为系统 server0 书写脚本/root/user.sh

运行脚本，可以判断用户输入的用户是否存在

如果存在，输出用户基本信息（id 用户名）

如果用户，不存在则创建用户，并输出用户创建成功

[root@server0 ~]# vim /root/user.sh

#!/bin/bash

read -p '请输入您测试的用户:' a

useradd $a &> /dev/null

if [ $? -eq 0 ];then

echo 用户$a创建成功

else

echo 用户$a已存在

id $a

fi

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

Samba服务基础,跨平台的共享，实现Linux与Windows数据的共享

• Samba 软件项目

– 用途:为客户机提供共享使用的文件夹

– 协议:SMB(TCP 139)、CIFS(TCP 445)

环境前提准备：

虚拟机Server0

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

虚拟机Desktop0：

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

**• 所需软件包:samba**

**• 系统服务:smb**

**• Samba用户 —— 专用来访问共享文件夹的用户**

**– 采用独立设置的密码**

**– 但需要提前建立同名的系统用户(可以不设密码)**

**– 与系统用户为同一个用户，但是不同密码**

**• 使用 pdbedit 管理工具**

**– 添加用户:pdbedit -a 用户名**

**– 查询用户:pdbedit -L [用户名]**

**– 删除用户:pdbedit -x 用户名**

**• 修改 /etc/samba/smb.conf**

**[自定共享名]**

**path = 文件夹绝对路径**

; public = no|yes //默认no

; browseable = yes|no //默认yes

; read only = yes|no //默认yes

; write list = 用户1 .. .. //默认无

; valid users = 用户1 .. .. //默认任何用户

; hosts allow = 客户机地址 .. ..

; hosts deny = 客户机地址 .. ..

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

环境前提准备：

虚拟机Server0

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

虚拟机Desktop0：

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

**虚拟机Server0**

**1.安装软件包samba**

**2.创建Samba共享验证用户（Samba共享帐号）**

[root@server0 ~]# useradd -s /sbin/nologin harry

[root@server0 ~]# useradd -s /sbin/nologin kenji

[root@server0 ~]# useradd -s /sbin/nologin chihiro

[root@server0 ~]# pdbedit -a harry #将harry添加为Samba共享帐号

[root@server0 ~]# pdbedit -a kenji

[root@server0 ~]# pdbedit -a chihiro

[root@server0 ~]# pdbedit -L #列出所有的Samba共享帐号

**3.修改配置文件/etc/samba/smb.conf**

[root@server0 ~]# mkdir /common

[root@server0 ~]# echo 123 > /common/1.txt

[root@server0 ~]# ls /common/

[root@server0 ~]# vim /etc/samba/smb.conf

#补充vim命令模式：按G（大写）直接到全文最后

#此服务器必须是 STAFF 工作组的一个成员

workgroup = STAFF #修改工作组

[common] #共享名

path = /common #共享的实际路径

**4.重起smb服务**

[root@server0 ~]# systemctl restart smb

[root@server0 ~]# systemctl enable smb

ln -s '/usr/lib/systemd/system/smb.service' '/etc/systemd/system/multi-user.target.wants/smb.service'

**5.布尔值（功能的开关）**

**– 需要加 -P 选项才能实现永久设置**

[root@server0 ~]# getsebool -a | grep samba

[root@server0 ~]# setsebool samba\_export\_all\_ro on

[root@server0 ~]# getsebool -a | grep samba

**客户端访问服务端资源：**

**1.防火墙策略**

**2.SELinux策略**

**3.服务本身的访问控制**

**4.服务端本地目录权限**

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

客户端虚拟机Desktop0

mount更加科学方便的访问方式

**1.创建挂载点**

[root@desktop0 ~]# mkdir /mnt/nsd

[root@desktop0 ~]# ls /mnt/nsd

**2.安装软件包cifs-utils 识别samba共享的数据**

[root@desktop0 ~]# yum -y install cifs-utils

[root@desktop0 ~]# mount -o user=harry,pass=123 //172.25.0.11/common /mnt/nsd/

[root@desktop0 ~]# df -h

[root@desktop0 ~]# ls /mnt/nsd

**3.完成开机自动挂载**

**\_netdev:声明网络设备**

**在开机启动时，需要将网络服务部署完成，自己具备IP地址后进行挂载**

**[root@desktop0 /]# vim /etc/fstab**

**//172.25.0.11/common /mnt/nsd cifs defaults,user=harry,pass=123,\_netdev 0 0**

[root@desktop0 /]# df -h

[root@desktop0 /]# umount /mnt/nsd

[root@desktop0 /]# mount -a

[root@desktop0 /]# df -h

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

（了解方式）虚拟机Desktop0

**1.安装软件包samba-client**

**2.访问服务端**

**• 列出共享资源**

**– smbclient -L 服务器地址**

**• 连接到共享文件夹**

**– smbclient -U 用户名 //服务器地址/共享名**

[root@desktop0 ~]# smbclient -L 172.25.0.11

[root@desktop0 ~]# smbclient -U harry //172.25.0.11/common

Enter harry's password:

Domain=[STAFF] OS=[Unix] Server=[Samba 4.1.1]

smb: \>

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

读写Samba共享

虚拟机Server0：

**1.创建共享目录**

[root@server0 ~]# mkdir /devops

[root@server0 ~]# echo 123 > /devops/a.txt

[root@server0 ~]# ls /devops/

**2.修改配置文件/etc/samba/smb.conf**

**文件的最后添加**

**[devops]**

**path = /devops**

**write list = chihiro**  #允许chihiro可以写入

**3.重起smb服务**

**[root@server0 ~]# systemctl restart smb**

**4.修改SELinux布尔值,开放读写功能**

[root@server0 ~]# getsebool -a | grep samba

[root@server0 ~]# setsebool samba\_export\_all\_rw on

[root@server0 ~]# getsebool -a | grep samba

**5.修改本地权限**

[root@server0 ~]# setfacl -m u:chihiro:rwx /devops/

[root@server0 ~]# getfacl /devops/

虚拟机Desktop0：

**1.书写开机自动挂载**

**[root@desktop0 /]# vim /etc/fstab**

//172.25.0.11/devops /mnt/pub cifs defaults,user=chihiro,pass=123,\_netdev 0 0

[root@desktop0 /]# mkdir /mnt/pub

[root@desktop0 /]# mount -a

[root@desktop0 /]# df -h

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

multiuser机制

• SMB客户端的 multiuser 挂载技术(只为普通用户设计)

– 管理员只需要作一次挂载

– 客户端在访问挂载点时,若需要不同权限,可以临时

切换为新的共享用户(无需重新挂载)

挂载参数调整

– multiuser,提供对客户端多个用户身份的区分支持

– sec=ntlmssp,提供NT局域网管理安全支持

[root@desktop0 /]# vim /etc/fstab

//172.25.0.11/devops /mnt/pub cifs defaults,user=kenji,pass=123,\_netdev,multiuser,sec=ntlmssp 0 0

[root@desktop0 /]# umount /mnt/pub

[root@desktop0 /]# mount -a

[root@desktop0 /]# su - student

[student@desktop0 ~]$ cifscreds add -u chihiro 172.25.0.11

Password:

[student@desktop0 ~]$ touch /mnt/pub/6.txt

[student@desktop0 ~]$ ls /mnt/pub/

[student@desktop0 ~]$ exit

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

配置NFS共享,Linux与Linux平台之间的共享

• Network File System,网络文件系统

– 用途:为客户机提供共享使用的文件夹

– 协议:NFS(TCP/UDP 2049)、RPC(TCP/UDP 111)

• 所需软件包:nfs-utils

• 系统服务:nfs-server

虚拟机Server0：

**1.安装软件包nfs-utils**

[root@server0 ~]# rpm -q nfs-utils

nfs-utils-1.3.0-0.el7.x86\_64

[root@server0 ~]#

**2.创建共享目录**

[root@server0 ~]# mkdir /nsd

[root@server0 ~]# echo haha > /nsd/abc.txt

[root@server0 ~]# ls /nsd

3.**修改配置文件/etc/exports**

– 文件夹路径 客户机地址(权限)

[root@server0 ~]# vim /etc/exports

/nsd \*(ro)

**4.重起服务nfs-server**

[root@server0 ~]# systemctl restart nfs-server

[root@server0 ~]# systemctl enable nfs-server

ln -s '/usr/lib/systemd/system/nfs-server.service' '/etc/systemd/system/nfs.target.wants/nfs-server.service'

客户端虚拟机Desktop0

[root@desktop0 /]# vim /etc/fstab

172.25.0.11:/nsd /mnt/test nfs defaults,\_netdev 0 0

[root@desktop0 /]# mkdir /mnt/test

[root@desktop0 /]# mount -a

[root@desktop0 /]# df -h

**2018-07-13(第五天)**

教学环境介绍

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server

[root@room9pc01 ~]# rht-vmctl reset desktop

[root@room9pc01 ~]# gos

[root@server0 ~]#

[root@server0 ~]# yum clean all #清空Yum缓存

[root@server0 ~]# yum repolist

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

配置高级连接

IP地址作用：唯一标识网络中一台主机

IPv4地址：32个二进制数组成，点 分隔 4部分，最终用4个十进制数表示

IPv6地址：

128个二进制数组成，冒号 分隔 8部分，每部分16二进制数，最终用4个16进制数表示

– 每段内连续的前置 0 可省略、连续的多个 : 可简化为 ::

**为两个虚拟机的接口 eth0 配置下列 IPv6 地址**

**– server0 上的地址应该是 2003:ac18::305/64**

[root@server0 ~]# nmcli connection modify 'System eth0' ipv6.method manual ipv6.addresses '2003:ac18::305/64' connection.autoconnect yes

[root@server0 ~]# nmcli connection up 'System eth0'

[root@server0 ~]# ifconfig | head -4

[root@server0 ~]# ping6 2003:ac18::305

配置聚合连接(链路聚合 网卡绑定 网卡组队)

解决网卡单点故障（热备份结构）

eth1 eth2

team0(虚拟网卡)192.168.1.1

**制作链路聚合**

**1.创建虚拟网卡** team0,再开一个新的终端参考 man teamd.conf

[root@server0 ~]# nmcli connection add type team

con-name team0 ifname team0 autoconnect yes

config '{"runner": {"name": "activebackup"}}'

[root@server0 ~]# ifconfig

nmcli connection 添加 类型 team（组队）

配置文件名 team0 网卡设备名 team0 每次开机自动启用

工作模式的配置 热备份方式工作模式

**2.添加成员**

[root@server0 ~]# nmcli connection add type team-slave

con-name team0-1 ifname eth1 master team0

[root@server0 ~]# nmcli connection add type team-slave

con-name team0-2 ifname eth2 master team0

nmcli connection 添加 类型 组队中成员

配置文件 team0-1 网卡设备 eth1 主设备为 team0

**3.为team0配置IP地址**

[root@server0 ~]# nmcli connection modify 'team0'

ipv4.method manual ipv4.addresses '192.168.1.1/24'

connection.autoconnect yes

**4.激活所有配置**

[root@server0 ~]# nmcli connection up team0

[root@server0 ~]# nmcli connection up team0-1

[root@server0 ~]# nmcli connection up team0-2

[root@server0 ~]# ifconfig

5.**专业查看team信息**

[root@server0 ~]# teamdctl team0 state

[root@server0 ~]# ifconfig eth1 down

[root@server0 ~]# teamdctl team0 state

**删除配置**

[root@server0 ~]# nmcli connection delete team0

[root@server0 ~]# nmcli connection delete team0-1

[root@server0 ~]# nmcli connection delete team0-2

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

**案例4:编写一个判断脚本**

在 server0 上创建 /root/foo.sh 脚本

1)当运行/root/foo.sh redhat,输出为fedora

2)当运行/root/foo.sh fedora,输出为redhat

3)当没有任何参数或者参数不是 redhat 或者

fedora时,其错误输出产生以下信息:

/root/foo.sh redhat|fedora

>&2:将正确输出信息，修改为错误输出

exit 3:退出脚本状态返回值为3

[root@server0 ~]# vim /root/foo.sh

#!/bin/bash

if [ $# -eq 0 ];then

echo '/root/foo.sh redhat|fedora' >&2

exit 3

elif [ $1 == redhat ];then

echo fedora

elif [ $1 == fedora ];then

echo redhat

else

echo '/root/foo.sh redhat|fedora' >&2

exit 4

fi

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

**案例5:编写一个批量添加用户脚本**

**在 server0 上创建 /root/batchusers 脚本**

1)此脚本要求提供用户名列表文件作为参数

2)如果没有提供参数,此脚本应该给出提示

Usage: /root/batchusers,退出并返回相应值

3)如果提供一个不存在的文件,

此脚本应该给出提示 Input file not found,退出并返回相应值

4)新用户的登录Shell为 /bin/false,无需设置密码

5)用户列表测试文件:

[root@server0 ~]# vim /root/user.txt

natasha

harry

chihiro

kenji

[root@server0 ~]# vim /root/batchusers

#!/bin/bash

if [ $# -eq 0 ];then

echo 'Usage: /root/batchusers' >&2

exit 2

elif [ -f $1 ];then

for i in `cat $1`

do

useradd -s /bin/false $i

echo $i创建成功

done

else

echo 'Input file not found' >&2

exit 3

fi

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

**2018-07-14(第六天)**

教学环境介绍

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server

[root@room9pc01 ~]# rht-vmctl reset desktop

[root@room9pc01 ~]# gos

[root@server0 ~]#

[root@server0 ~]# yum clean all #清空Yum缓存

[root@server0 ~]# yum repolist

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

iSCSI网络磁盘 ,默认端口 3260

**• Internet SCSI,网际SCSI接口**

– 一种基于C/S架构的虚拟磁盘技术

– 服务器提供磁盘空间,客户机连接并当成本地磁盘使用

达外冰箱------->纸的包装------->制作 木箱子------>发送

**• backstore,后端存储**

**– 对应到服务端提供实际存储空间的设备,需要起一个管理名称**

**• target,磁盘组**

**– 是客户端的访问目标,作为一个框架,由多个lun组成**

**• lun,逻辑单元**

**– 每一个lun需要关联到某一个后端存储设备,在客户端**

**会视为一块虚拟硬盘**

**• ISCSI Qualified Name 名称规范（iqn）**

**– iqn.yyyy-mm.倒序域名:自定义标识**

环境前提：

修改两台虚拟机的防火墙：

[root@server0 ~]# firewall-cmd --set-default-zone=trusted

[root@desktop0 ~]# firewall-cmd --set-default-zone=trusted

虚拟机Server0

1.划分新的分区,该主分区使用的空间，为硬盘所有的空间

[root@server0 ~]# fdisk /dev/vdb

按n创建新的分区

[root@server0 ~]# lsblk

2.安装软件包targetcli

3.运行命令进行配置

[root@server0 ~]# targetcli

1.建立后端存储(/dev/vdb1类似于冰箱----->nsd类似于纸箱子)

/> backstores/block create name=nsd dev=/dev/vdb1

/> ls

2.建立target磁盘组（木质箱子）

/> iscsi/ create iqn.2018-07.example.com:server0

/> ls

3.关联lun

/> iscsi/iqn.2018-07.example.com:server0/tpg1/luns

create /backstores/block/nsd

/> ls

4.设置访问控制,设置客户端声称的名字

/> iscsi/iqn.2018-07.example.com:server0/tpg1/acls

create iqn.2018-07.example.com:desktop0

/> ls

客户端在访问服务端共享存储时声称的名字

5.开放网络接口

/> iscsi/iqn.2018-07.example.com:server0/tpg1/portals

create 172.25.0.11

/> ls

/> exit

重起服务target

[root@server0 ~]# systemctl restart target

[root@server0 ~]# systemctl enable target

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

虚拟机Desktop0

1.安装客户端访问程序

# yum repolist #生成Yum缓存

# yum -y install iscsi-initiator-utils.i686

# yum -y install isc(tab)

[root@desktop0 ~]# rpm -q iscsi-initiator-utils

iscsi-initiator-utils-6.2.0.873-21.el7.x86\_64

iscsi-initiator-utils-6.2.0.873-21.el7.i686

2.修改配置文件，指定客户端生成的名字

[root@desktop0 ~]# vim /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2018-07.example.com:desktop0

[root@desktop0 ~]# systemctl restart iscsid //刷新IQN标识

Warning: Unit file of iscsid.service changed on disk, 'systemctl daemon-reload' recommended.

[root@desktop0 ~]# systemctl daemon-reload

[root@desktop0 ~]# systemctl restart iscsid

3.发现服务端共享存储 参考man iscsiadm 按G(大写)到全文最后

字体变大：Ctrl shift +

字体变小：Ctrl -

[root@desktop0 ~]# iscsiadm --mode discoverydb --type sendtargets --portal 172.25.0.11 --discover

4.加载共享存储

[root@desktop0 ~]# lsblk

[root@desktop0 ~]# systemctl restart iscsi #客户端服务

[root@desktop0 ~]# lsblk

[root@desktop0 ~]# systemctl enable iscsi

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

数据库服务基础

什么是数据库:存放数据的仓库

• 禁止监听,只服务于本机

[root@server0 ~]# vim /etc/my.cnf #数据库主配置文件

[mysqld]

skip-networking #跳过网络监听

**1.安装数据库软件**

**虚拟机server0**

**– mariadb-server:提供服务端有关的系统程序,默认端口 3306**

**[root@server0 ~]# yum -y install mariadb-server**

2.启动数据库服务

[root@server0 ~]# systemctl restart mariadb

[root@server0 ~]# systemctl enable mariadb

二、数据库基本操作

1.使用 mysql 命令，登入交互界面,实现数据库管理操作

[root@server0 ~]# mysql

> show databases; #查看所有库

> create database nsd1806; #创建库nsd1806

> show databases;

> drop database nsd1806; #删除库nsd1806

> show databases;

> exit

Bye

2.为数据库管理员设置密码

数据库管理员root：数据库MariaDB最高权限用户，mysql库中user表

系统管理员root：Linux系统最高权限用户，/etc/passwd

• 为数据库账号修改密码

– mysqladmin [-u用户名] [-p[旧密码]] password '新密码'

[root@server0 ~]# mysqladmin -u root password '123'

[root@server0 ~]# mysql -u root -p #交互式设置密码

Enter password:

[root@server0 ~]# mysql -u root -p123 #非交互式设置密码

3.导入数据到数据库

[root@server0 ~]# mysql -u root -p123

MariaDB [(none)]> create database nsd; #创建库nsd

MariaDB [(none)]> show databases; #查看所有库

MariaDB [(none)]> exit

[root@server0 ~]# wget http://classroom.example.com/pub/materials/users.sql

将数据导入到库nsd中

[root@server0 ~]# mysql -u root -p123 nsd < users.sql

[root@server0 ~]# mysql -u root -p123

MariaDB [(none)]> use nsd; #进入库nsd

MariaDB [nsd]> show tables; #查看当前库都有那些表格

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

4.表格操作： 增(insert) 删(delete) 改(update) 查(select)

表记录的查询操作：

[root@server0 ~]# mysql -u root -p123

MariaDB [(none)]> use nsd;

MariaDB [nsd]> show tables;

MariaDB [nsd]> select 表字段 from 表名字;

MariaDB [nsd]> select \* from base;

MariaDB [nsd]> select \* from location;

MariaDB [nsd]> select id,name from base;

MariaDB [nsd]> select name,password from base;

MariaDB [nsd]> desc 表名字; #查看表结构

MariaDB [nsd]> use mysql;

MariaDB [nsd]> show tables;

MariaDB [mysql]> desc user;

MariaDB [mysql]> select user,host,password from user;

MariaDB [mysql]> select \* from nsd.base;

有条件的查询：

> use nsd;

> select \* from base where name='tom';

> select \* from base where name='barbara';

> select \* from base where password='123';

> select id,name from base where password='123';

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

数据库的授权

– 除了 root 用户,此nsd数据库只能被用户 lisi 查询,

此用户的密码为123

– grant 权限列表 on 数据库名.表名 to 用户名@客户机地址

identified by '密码';

当lisi从localhost本地进行登陆，输入密码123，将会获得nsd库中所有表的读取权限。

[root@server0 ~]# mysql -u root -p123

> grant select on nsd.\* to lisi@localhost identified

by '123';

> select user,host,password from mysql.user;

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

案例5:使用数据库查询

2. 在系统 server0 上使用数据库 nsd,并使用相

应的 SQL 查询以回答下列问题:

1)密码是 solicitous 的人的名字?

[root@server0 ~]# mysql -u root -p123

> use nsd;

> select \* from base where password='solicitous';

> select name from base where password='solicitous';

2) 有多少人的姓名是 Barbara 同时居住在 Sunnyvale?

> select \* from base,location

where base.name='Barbara' and

location.city='Sunnyvale' and base.id=location.id ;

> select count(\*) from base,location

where base.name='Barbara' and

location.city='Sunnyvale' and base.id=location.id ;

> insert base values('6','Barbara','789');

> select \* from base;

> insert location values('6','Sunnyvale');

> select \* from location ;

1. 禁止空密码root用户访问 mariadb 数据库

[root@server0 ~]# mysql -u root -p123

MariaDB [(none)]> use mysql;

MariaDB [mysql]> select user,host,password from user;

删除表中，密码为空的记录

MariaDB [mysql]> delete from user where password='';

MariaDB [mysql]> select user,host,password from user;

MariaDB [mysql]> flush privileges; #刷新策略

MariaDB [mysql]> exit

Bye

[root@server0 ~]# mysql -u root -h server0.example.com

ERROR 1130 (HY000): Host 'server0.example.com' is not allowed to connect to this MariaDB server

登陆不成功为正确

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

**2018-07-15(第七天)**

教学环境介绍

[root@room9pc01 ~]# rht-vmctl reset classroom

[root@room9pc01 ~]# rht-vmctl reset server

[root@room9pc01 ~]# rht-vmctl reset desktop

环境前提：

修改两台虚拟机的防火墙：

[root@server0 ~]# firewall-cmd --set-default-zone=trusted

[root@desktop0 ~]# firewall-cmd --set-default-zone=trusted

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

HTTP服务基础

• 基于 B/S (Browser/Server)架构的网页服务

– 服务端提供网页

– 浏览器下载并显示网页

• Hyper Text Markup Language,超文本标记语言(html)

• Hyper Text Transfer Protocol,超文本传输协议(http)

– Listen:监听地址:端口(80)

– ServerName:本站点注册的DNS名称(空缺)

– DocumentRoot:网页根目录(/var/www/html)

– DirectoryIndex:起始页/首页文件名(index.html)

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

一、安全的Web服务器（https安全的超文本传输协议 默认端口 443）

回顾虚拟Web主机：

虚拟机Server0

1.安装httpd软件

# echo '<h1>NSD1806 Web1' > /var/www/html/index.html

# cat /var/www/html/index.html

# cat /etc/httpd/conf.d/nsd01.conf

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /var/www/html

</VirtualHost>

# systemctl restart httpd

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

构建安全加密的Web服务器

一、基础设施

– 公钥:主要用来加密数据

– 私钥:主要用来解密数据(与相应的公钥匹配)

– 数字证书:证明拥有者的合法性/权威性(单位名称、

有效期、公钥、颁发机构及签名、......)

– Certificate Authority,数字证书授权中心:负责证书

的申请/审核/颁发/鉴定/撤销等管理工作

虚拟机Server0

1.部署网站证书(营业执照)

# cd /etc/pki/tls/certs/

# wget http://classroom.example.com/pub/tls/certs/server0.crt

# ls

2.部署根证书(公安局信息)

# cd /etc/pki/tls/certs/

# wget http://classroom.example.com/pub/example-ca.crt

# ls

3.部署私钥(用于解密数据)

# cd /etc/pki/tls/private/

# wget http://classroom.example.com/pub/tls/private/server0.key

# ls

4.安装mod\_ssl软件包，提供安全支持

# yum -y install mod\_ssl

5.修改配置文件/etc/httpd/conf.d/ssl.conf

[root@server0 /]# vim /etc/httpd/conf.d/ssl.conf

补充：vim末行模式 输入 :set nu 实现添加行号

59行 DocumentRoot "/var/www/html"

60行 ServerName www0.example.com:443

指定网站证书位置

100行 SSLCertificateFile /etc/pki/tls/certs/server0.crt

指定私钥位置

107行 SSLCertificateKeyFile /etc/pki/tls/private/server0.key

指定根证书的位置

122行 SSLCACertificateFile /etc/pki/tls/certs/example-ca.crt

6.重起httpd服务

[root@server0 /]# systemctl restart httpd

7.验证

[root@desktop0 ~]# firefox https://www0.example.com

选择 我了解风险------->添加例外------->确认安全例外

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

基础邮件服务

• 电子邮件服务器的基本功能

– 为用户提供电子邮箱存储空间(用户名@邮件域名)

– 处理用户发出的邮件（SMTP） —— 传递给收件服务器

– 处理用户收到的邮件（pop3 IMAP）—— 投递到邮箱

虚拟机Server0

1.安装postfix软件包

[root@server0 ~]# yum -y install postfix

[root@server0 ~]# rpm -q postfix

2.修改配置文件/etc/postfix/main.cf

[root@server0 ~]# vim /etc/postfix/main.cf

补充：vim末行模式 输入 :set nu 实现添加行号

99行 myorigin = server0.example.com #默认补全的域名后缀

116行 inet\_interfaces = all #允许所有人使用邮件服务

164行 mydestination = server0.example.com #判断为本域邮件

3.重起postfix服务

[root@server0 ~]# systemctl restart postfix

4.创建本地用户进行收发邮件

[root@server0 ~]# useradd yg

[root@server0 ~]# useradd xln

• mail 发信操作

– mail -s '邮件标题' -r 发件人 收件人[@收件域]...

• mail 收信操作

– mail [-u 用户名]

[root@server0 ~]# mail -s '葵花宝典' -r yg xln

haha xixi hehelele

. #提交邮件信息

[root@server0 ~]# mail -u xln

Heirloom Mail version 12.5 7/5/10. Type ? for help.

"/var/mail/xln": 1 message 1 new

>N 1 yg@server0.example.c Tue Jul 17 14:25 19/631

& 1

& q #退出

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

parted分区工具，划分分区

GPT：大容量的划分，2T以上，最多支持128个主分区

[root@server0 ~]# parted /dev/vdb

(parted) mktable gpt #指定分区模式

(parted) print #输出分区表信息

(parted) mkpart #划分新的分区

分区名称？ []? nsd #分区的名字随便起

文件系统类型？ [ext2]? ext4 #分区的文件系统，随便写无实际作用

起始点？ 0

结束点？ 2G

忽略/Ignore/放弃/Cancel? Ignore #选择忽略

(parted) unit GB #选择显示容量的单位为GB

(parted) print

(parted) mkpart

分区名称？ []? nsd

文件系统类型？ [ext2]? ext4

起始点？ 2G

结束点？ 4G

(parted) print

(parted) quit

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

什么是交换空间

• 相当于虚拟内存,

– 当物理内存不够用时,使用磁盘空间来模拟内存

– 在一定程度上缓解内存不足的问题

– 交换分区:以空闲分区充当的交换空间

一、 创建交换空间

1.前提要有空闲的分区

2.首先格式化交换文件系统

[root@server0 ~]# mkswap /dev/vdb1

[root@server0 ~]# mkswap /dev/vdb2

正在设置交换空间版本 1，大小 = 1953104 KiB

无标签，UUID=cdf9c261-ccee-42a4-be59-d2859e9e81f6

[root@server0 ~]# blkid /dev/vdb1

3. 启用交换分区

[root@server0 ~]# swapon /dev/vdb1 #启用/dev/vdb1交换分区

[root@server0 ~]# swapon -s #查看交换空间的组成 成员

[root@server0 ~]# swapon /dev/vdb2

[root@server0 ~]# swapon -s

[root@server0 ~]# swapoff /dev/vdb2 #停用交换分区

[root@server0 ~]# swapon -s

4.完成开机自动启用，交换分区

[root@server0 ~]# vim /etc/fstab

/dev/vdb1 swap swap defaults 0 0

/dev/vdb2 swap swap defaults 0 0

[root@server0 ~]# swapoff /dev/vdb1

[root@server0 ~]# swapon -s

[root@server0 ~]# swapon -a #检测交换分区是否书写正确

[root@server0 ~]# swapon -s

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

**三、Serices**

**2018-07-18(第一天)**

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

目录结构

• 认识Linux的目录层次:

– man hier

主要用途

/boot 存放系统引导必需的文件,包括内核、启动配置

/bin、/sbin 存放各种命令程序

/dev 存放硬盘、键盘、鼠标、光驱等各种设备文件

/etc 存放Linux系统及各种程序的配置文件

/root、/home/用户名 分别是管理员root、普通用户的默认家目录

/var 存放日志文件、邮箱目录等经常变化的文件

/proc 存放内存中的映射数据,不占用磁盘

/tmp 存放系统运行过程中使用的一些临时文件

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

搭建教学环境

一、真机搭建FTP服务

1.安装vsftpd软件

[root@room9pc01 ~]# rpm -q vsftpd

vsftpd-3.0.2-22.el7.x86\_64

2.查看服务启动

[root@room9pc01 ~]# systemctl status vsftpd

[root@room9pc01 ~]# systemctl restart vsftpd

[root@room9pc01 ~]# systemctl enable vsftpd

3.共享光盘所有内容

默认共享：/var/ftp/

服务端: 1.众多的软件包 2.仓库清单文件 3.共享的服务

[root@room9pc01 ~]# vim /etc/fstab

/var/lib/libvirt/images/iso/rhel-server-7.4-x86\_64-dvd.iso

/var/ftp/rhel7 iso9660 defaults 0 0

[root@room9pc01 ~]# ls /var/ftp/rhel7

[root@room9pc01 ~]# firefox ftp://127.0.0.1/rhel7

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

[root@room9pc01 ~]# clone-vm7 #快速产生虚拟机

Enter VM number: 1

[root@room9pc01 ~]# clone-vm7

Enter VM number: 2

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

利用root进行登陆，密码为123456

利用真机进行远程管理：

虚拟机A：

1.主机名：svr7.tedu.cn

2.配置eth0网卡IP地址：192.168.4.7/24

3.配置Yum仓库（以真机作为网络Yum源）

虚拟机B：

1.主机名：pc207.tedu.cn

2.配置eth0网卡IP地址：192.168.4.207/24

3.配置Yum仓库（以真机作为网络Yum源）

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

虚拟机A

[root@svr7 ~]# vim /etc/yum.repos.d/rhel7.repo

[rhel7]

name=rhel7.4

baseurl=ftp://192.168.4.254/rhel7

enabled=1

gpgcheck=0

[root@svr7 ~]# yum repolist

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

补充内容：将本地的数据，传递给网络中其他主机

[root@svr7 ~]# scp /etc/yum.repos.d/rhel7.repo

root@192.168.4.207:/etc/yum.repos.d/

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

权限的数值表示

• 权限的数值化

– 基本权限:r = 4,w = 2,x = 1

– 附加权限:SUID = 4,SGID = 2,Sticky Bit = 1

[root@svr7 ~]# mkdir /nsd01

[root@svr7 ~]# ls -ld /nsd01

[root@svr7 ~]# chmod 700 /nsd01

[root@svr7 ~]# ls -ld /nsd01

[root@svr7 ~]# chmod 007 /nsd01

[root@svr7 ~]# ls -ld /nsd01

[root@svr7 ~]# chmod 750 /nsd01

[root@svr7 ~]# ls -ld /nsd01

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

历史命令

• 管理/调用曾经执行过的命令

– history:查看历史命令列表

– history -c:清空历史命令

– !n:执行命令历史中的第n条命令

– !str:执行最近一次以str开头的历史命令

• 调整历史命令的数量

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

HISTSIZE=1000 //默认记录1000条

[root@svr7 ~]# history

[root@svr7 ~]# history -c #清空历史命令

[root@svr7 ~]# history #查看历史命令

[root@svr7 ~]# cat /etc/redhat-release

[root@svr7 ~]# uname -r

[root@svr7 ~]# hostname

[root@svr7 ~]# history

[root@svr7 ~]# !cat #执行最近一条以cat开头的历史命令

[root@svr7 ~]# !h

[root@svr7 ~]# !u

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

实用小命令工具

• du,统计文件的占用空间

– du [选项]... [目录或文件]...

– -s:只统计每个参数所占用的总空间大小

– -h:提供易读容量单位(K、M等)\

[root@svr7 ~]# du -sh /boot/ /etc/pki/

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

• date,查看/调整系统日期时间

– date +%F、date +%R

– date +"%Y-%m-%d %H:%M:%S"

– date -s "yyyy-mm-dd HH:MM:SS"

[root@svr7 ~]# date +%R

[root@svr7 ~]# date +%F

[root@svr7 ~]# date +%Y

[root@svr7 ~]# date +%m

[root@svr7 ~]# date +%d

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

Yum排错：

1. 格式是否正确

[root@svr7 ~]# cat /etc/yum.repos.d/rhel7.repo

2.查看是否有光盘的内容

[root@svr7 ~]# firefox ftp://192.168.4.254/rhel7

没有：1.是否ping通 2.到真机vsftpd服务是否开启

3.ls /var/ftp/rhel7----》没有挂载------》/etc/fstab

4.真机开启防火墙

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

创建自定义Yum源

在真机上操作

1.将真机上tools.tar.gz 传递到虚拟机A

# cd /root/桌面

# scp tools.tar.gz root@192.168.4.7:/root/

2.在虚拟机A上验证

# ls /root

3.在虚拟机A上,解包

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

[root@svr7 ~]# ls /

[root@svr7 ~]# ls /tools/

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

4.生成 仓库数据文件

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

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

5.书写Yum客户端配置文件

[root@svr7 ~]# vim /etc/yum.repos.d/rhel7.repo

[myrpm]

name=my rpm

baseurl=file:///tools/other #指定本地为Yum服务端

enabled=1

gpgcheck=0

[root@svr7 ~]# yum repolist

[root@svr7 ~]# rpm -q sl

sl-5.02-1.el7.x86\_64

[root@svr7 ~]# rpm -ql sl #查询软件包安装的清单

[root@svr7 ~]# oneko & #放入后台

[root@svr7 ~]# killall oneko #杀死程序

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

实用小命令工具(续2)

• ln,创建软连接(快捷方式)

– 软连接 --> 原始文档 --> i节点 --> 文档数据

– ln -s 原始文件或目录 软连接文件

若原始文件或目录被删除,连接文件将失效

软连接可存放在不同分区/文件系统

• ln,创建硬连接(快捷方式)

– 硬连接 --> i节点 --> 文档数据

– ln 原始文件 硬连接文件

若原始文件被删除,连接文件仍可用

硬连接与原始文件必须在同一分区/文件系统

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

制作快捷方式

[root@svr7 ~]# cat /etc/redhat-release

[root@svr7 ~]# ln -s /etc/redhat-release /

[root@svr7 ~]# ls /

[root@svr7 ~]# ln -s /etc/redhat-release /red

[root@svr7 ~]# ls /

[root@svr7 ~]# ls -l /red

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

i节点（编号） :标识硬盘一块存储的区域

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

[root@svr7 ~]# ls /opt/

[root@svr7 ~]# ln -s /opt/1.txt /opt/A #软链接，对源文件

[root@svr7 ~]# ls /opt/

[root@svr7 ~]# ln /opt/1.txt /opt/B #硬链接，对i节点

[root@svr7 ~]# ls -i /opt/

[root@svr7 ~]# rm -rf /opt/1.txt

[root@svr7 ~]# cat /opt/A

[root@svr7 ~]# cat /opt/B

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

查看命令的帮助

• 通常 会提供“--help”帮助选项

• man,格式化手册阅读工具

5 配置文件帮助信息

[root@svr7 ~]# man passwd

[root@svr7 ~]# man 5 passwd

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

zip归档工具，跨平台的压缩归档工具

– zip [-r] 备份文件.zip 被归档的文档...

[root@svr7 ~]# zip -r /opt/file.zip /home/ /etc/passwd

[root@svr7 ~]# zip -r /opt/test.zip /usr/local

[root@svr7 ~]# ls /opt

• 释放归档+解压操作

– unzip 备份文件.zip [-d 目标文件夹]

[root@svr7 ~]# ls /mnt/

[root@svr7 ~]# unzip /opt/file.zip -d /mnt/

[root@svr7 ~]# ls /mnt/

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

vim编辑技巧

命令模式操作

光标跳转

Home 键 或 ^、 数字 0 跳转到行首

End 键 或“$”键 跳转到行尾

1G 或 gg 跳转到文件的首行

G 跳转到文件的末尾行

10G 跳转到文件的第10行

复制/粘贴/删除

yy、#yy 复制光标处的一行、#行

p、P 粘贴到光标处之后、之前

x 或 Delete键 删除光标处的单个字符

dd、#dd 删除光标处的一行、#行

d^ 从光标处之前删除至行首

d$ 从光标处删除到行尾

C(大写) 从光标处删除到行尾，进入编辑模式

u 撤销

查找/撤销/保存

/word 向后查找字符串“word”

n、N 跳至后/前一个结果

u 撤销最近的一次操作

Ctrl + r 取消前一次撤销操作(反撤销)

ZZ(大写） 保存修改并退出

末行模式操作

保存/退出/文件操作

:q! 放弃已有更改后强制退出

:r /etc/filesystems 读入其他文件内容

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

[root@svr7 ~]# echo xixi > /opt/2.txt

[root@svr7 ~]# vim /opt/1.txt

:r /opt/2.txt #读入/opt/2.txt文件内容到当前文本

:r /etc/passwd

字符串替换

:s/root/admin 光标所在行，替换当前行第一个“root”

:s/root/admin/g 光标所在行，替换当前行所有的“root”

:1,10 s/root/admin/g 替换第1-10行所有的“root”

:% s/root/admin/g 替换全文的所有的“root”

开关参数的控制

:set nu|nonu 显示/不显示行号

:set ai|noai 启用/关闭自动缩进(每次光标以上一行为基础，对齐)

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

**2018-07-20(第二天)**

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

环境准备：检查Yum是否可用

虚拟机A：

[root@svr7 ~]# yum clean all #清空Yum缓存

[root@svr7 ~]# yum repolist

虚拟机B：

[root@pc207 ~]# yum clean all #清空Yum缓存

[root@pc207 ~]# yum repolist

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

基础DNS服务

• DNS服务器的功能

– 正向解析:根据注册的域名查找其对应的IP地址

– 反向解析:根据IP地址查找对应的注册域名,不常用

域名体系：

所有的域名，都要以 点 结尾(称为 根域名)

根 .

一级域名： .cn .kr .hk .tw .jp .us .com .net .org ....

二级域名： .com.cn .edu.cn .net.cn .gov.cn .mil.cn ....

三级域名: .dawai.com.cn .nb.com.cn .haxi.com.cn .dc.com.cn

完整主机名: www.dawai.com.cn ftp.dawai.com.cn tts.dawei.com.cn

• Full Qualified Domain Name,完全合格主机名(FQDN)

– = 站点名.域名后缀

– = 站点名. .. .. .二级域.一级域

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

BIND域名服务

• BIND(Berkeley Internet Name Daemon)

– 伯克利 Internet 域名服务

– 官方站点:https://www.isc.org/

• 主配置文件:/etc/named.conf #设置本机负责解析的域名

• 地址库文件:/var/named/ #所有的完整的主机名与IP对应关系

– 系统服务:named

– 默认端口:TCP/UDP 53

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

**搭建基本的DNS服务**

**虚拟机A：**

**1.安装软件包**

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

bind-9.9.4-29.el7.x86\_64 #域名服务包

bind-chroot-9.9.4-29.el7.x86\_64 #提供虚拟根支持(牢笼政策)

**2.修改主配置文件/etc/named.conf**

[root@svr7 ~]# cp /etc/named.conf /root/name.bak

[root@svr7 ~]# vim /etc/named.conf

options {

directory "/var/named"; #指定地址库文件存放路径

};

zone "tedu.cn" IN { #指定本DNS服务器负责解析的域名

type master; #指定本机为权威DNS服务器(主DNS)

file "tedu.cn.zone"; #指定地址库文件的名字

};

**3.建立地址库文件/var/named/tedu.cn.zone**

# cd /var/named

# cp -p named.localhost tedu.cn.zone #保持属性不变，进行复制

# vim tedu.cn.zone

所有的域名必须以 "." 作为结尾

没有以 "." 作为结尾，则默认补全地址库文件负责的域名

tedu.cn. NS svr7

svr7 A 192.168.4.7

www A 1.2.3.4

ftp A 5.6.7.8

虚拟机B：验证

1.指定DNS服务器

[root@pc207 ~]# echo nameserver 192.168.4.7 > /etc/resolv.conf

[root@pc207 ~]# nslookup www.tedu.cn

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

**多区域DNS服务器**

**虚拟机A：**

**1.修改主配置文件/etc/named.conf**

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

**2.建立地址库文件/var/named/qq.com.zone**

# cd /var/named/

# cp -p tedu.cn.zone qq.com.zone

# vim qq.com.zone

qq.com. NS svr7

svr7 A 192.168.4.7

www A 2.2.2.2

ftp A 3.3.3.3

**3.重起named服务**

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

特殊的解析记录

一、DNS负载均衡(解析结果的轮询)

[root@svr7 /]# vim /var/named/qq.com.zone

qq.com. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.1

www A 192.168.4.2

www A 192.168.4.3

www A 192.168.4.4

ftp A 3.3.3.3

[root@svr7 /]# systemctl restart named

[root@pc207 ~]# ping www.qq.com

[root@pc207 ~]# ping www.qq.com

**二、泛域名解析**

虚拟机A：

[root@svr7 /]# vim /var/named/qq.com.zone

qq.com. NS svr7

svr7 A 192.168.4.7

........

\* A 10.11.12.13

qq.com. A 14.15.16.17

[root@svr7 /]# systemctl restart named

虚拟机B：

[root@pc207 /]# nslookup www.qq.com

[root@pc207 /]# nslookup wwww.qq.com

[root@pc207 /]# nslookup haha.qq.com

[root@pc207 /]# nslookup qq.com

**三、有规律的泛域名解析**

pc1.qq.com-------》192.168.10.1

pc2.qq.com-------》192.168.10.2

pc3.qq.com-------》192.168.10.3

......

pc50.qq.com-------》192.168.10.50

DNS服务内置的变量： $GENERATE 可以产生连续的数字

虚拟机A：

[root@svr7 /]# vim /var/named/qq.com.zone

$GENERATE 1-50 pc$ A 192.168.10.$

#产生连续范围的数字1至50 ，其中$代表1至50 范围内的所有数字

[root@svr7 /]# systemctl restart named

虚拟机B：

[root@pc207 /]# nslookup pc1.qq.com

[root@pc207 /]# nslookup pc3.qq.com

[root@pc207 /]# nslookup pc28.qq.com

[root@pc207 /]# nslookup pc50.qq.com

**四、解析记录的别名**

虚拟机A：

[root@svr7 /]# vim /var/named/qq.com.zone

**tts CNAME ftp**

#tts解析结果与ftp解析结果相同

[root@svr7 /]# systemctl restart named

虚拟机B：

[root@pc207 /]# nslookup tts.qq.com

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

**DNS子域授权**

父域: www.tedu.cn

子域: www.bj.tedu.cn

虚拟机A 负责解析 父域 tedu.cn

虚拟机B 负责解析 子域 bj.tedu.cn

虚拟机A 可以解析 子域 bj.tedu.cn

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

**虚拟机B：**

**1.安装软件包 bind-chroot bind**

**2.修改主配置文件/etc/named.conf**

[root@pc207 ~]# cp /etc/named.conf /root/name.bak

[root@pc207 ~]# vim /etc/named.conf

options {

directory "/var/named";

};

zone "bj.tedu.cn" IN {

type master;

file "bj.tedu.cn.zone";

};

**3.建立地址库文件/var/named/bj.tedu.cn.zone**

# cp -p /var/named/named.localhost /var/named/bj.tedu.cn.zone

# vim /var/named/bj.tedu.cn.zone

bj.tedu.cn. NS pc207

pc207 A 192.168.4.207

www A 6.6.6.6

[root@pc207 /]# systemctl restart named

[root@pc207 /]# nslookup www.bj.tedu.cn 192.168.4.207

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

**虚拟机A 可以解析 子域 bj.tedu.cn （子域授权）**

**虚拟机A：**

**1.修改地址库文件**

**[root@svr7 /]# vim /var/named/tedu.cn.zone**

tedu.cn. NS svr7

bj.tedu.cn. NS pc207

svr7 A 192.168.4.7

pc207 A 192.168.4.207

www A 1.2.3.4

ftp A 5.6.7.8

[root@svr7 /]# systemctl restart named

2.验证：

[root@pc207 /]# nslookup www.bj.tedu.cn 192.168.4.7

Server: 192.168.4.7

Address: 192.168.4.7#53

Non-authoritative answer: #非权威解答

Name: www.bj.tedu.cn

Address: 6.6.6.6

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

DNS查询的工作方式

递归解析：

客户端询问DNS服务器解析请求，DNS服务器与其他DNS服务器交互，最终把解析

结果带回来

options {

directory "/var/named";

**recursion no; #禁止递归**

};

迭代解析：首选DNS服务器与其他DNS服务器交互过程

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

主机映射文件/etc/hosts(只为本机提供DNS域名解析的功能)

[root@svr7 /]# vim /etc/hosts

192.168.4.110 www.sina.com

[root@svr7 /]# ping www.sina.com

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

总结：客户端DNS解析域名过程

1.客户端查询/etc/hosts

2.客户端查询/etc/resolv.conf 查看DNS服务器

3.DNS服务器进行递归查询与迭代查询

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

缓存DNS，加速解析效率

**真机搭建Yum仓库**

**1.真机挂载光盘**

# cat /etc/redhat-release

# mkdir /dvd

# mount /iso/CentOS-7-x86\_64-DVD-1708.iso /dvd/

# ls /dvd

2.真机书写客户端配置文件

# cd /etc/yum.repos.d/

# ls

# mkdir repo

# mv \*.repo repo #排除错误的文件干扰

# vim dvd.repo

[dvd]

name=CentOS7.4

baseurl=file:///dvd

enabled=1

gpgcheck=0

# yum -y install bind bind-chroot

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

**搭建缓存DNS服务器**

**1.查看真机DNS服务器IP地址**

[root@room9pc01 /]# cat /etc/resolv.conf

nameserver 172.40.1.10

**2.配置缓存DNS服务器**

**[root@room9pc01 /]# vim /etc/named.conf**

options {

directory "/var/named";

forwarders { 172.40.1.10; }; #转发给达内DNS服务器

};

**3.重起named服务**

**虚拟机A上验证：**

**[root@svr7 /]# nslookup www.qq.com 192.168.4.254**

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