# 目录

[ADMIN 5](#_Toc385081)

[开学典礼、云计算介绍、 5](#_Toc385082)

[ADMIN云计算系统管理: 5](#_Toc385083)

[ENGINEER云计算应用管理 5](#_Toc385084)

[SERVICES系统&服务管理进阶 5](#_Toc385085)

[教学环境 5](#_Toc385086)

[真机使用rht-vmctl辅助工具 6](#_Toc385087)

[虚拟机server： 6](#_Toc385088)

[虚拟机软件： 6](#_Toc385089)

[ADMIN01 7](#_Toc385090)

**[TCP/IP协议及配置](#_Toc385091)** [7](#_Toc385091)

**[IP地址(IP address)](#_Toc385092)** [7](#_Toc385092)

**[IP地址的分类](#_Toc385093)** [7](#_Toc385093)

[ADMIN02 9](#_Toc385094)

[一：Linux系统简介 9](#_Toc385095)

[一：RHEL系统与CentOS系统的渊源 9](#_Toc385096)

[二：RHEL7的根分区默认文件系统 9](#_Toc385097)

[三：Linux分区表示方式 10](#_Toc385098)

[二：Linux命令行基本操作 10](#_Toc385099)

[一、获取命令行界面的不同方法 10](#_Toc385100)

[二、简单命令行操作练习 11](#_Toc385101)

[三：Linux 的五个重启命令 12](#_Toc385102)

[1.shutdown 12](#_Toc385103)

[2.halt----最简单的关机命令 13](#_Toc385104)

[3.reboot 13](#_Toc385105)

[4.init 14](#_Toc385106)

[5.poweroff---立即关机 14](#_Toc385107)

[四：IP地址的概述 14](#_Toc385108)

[• IP地址的分类 14](#_Toc385109)

[ADMIN03 15](#_Toc385110)

[• Linux命令 15](#_Toc385111)

[1．清理编辑的命令基础技巧 15](#_Toc385112)

[2．mount挂载 15](#_Toc385113)

[mount挂载概念： 15](#_Toc385114)

[mount案例：挂载并访问光盘设备 16](#_Toc385115)

[3．ls列表及文档创建 17](#_Toc385116)

[ls — List格式 17](#_Toc385117)

[文档开头判断 17](#_Toc385118)

[4．目录管理 17](#_Toc385119)

[mkdir 创建目录 17](#_Toc385120)

[cd — Change Directory 17](#_Toc385121)

[4.1 目录管理案例 17](#_Toc385122)

[5．通配符使用 18](#_Toc385123)

[5.1 通配符案例 18](#_Toc385124)

[6．Vim编辑器 18](#_Toc385125)

[步骤一：使用ls命令列出指定的文件 19](#_Toc385126)

[步骤二：mkdir新建文档 19](#_Toc385127)

[7．复制、删除、移动 19](#_Toc385128)

[rm 删除 19](#_Toc385129)

[mv 移动/改名 20](#_Toc385130)

[cp 复制 20](#_Toc385131)

[8．查看大文本内容操作 21](#_Toc385132)

[• less分屏阅读工具 21](#_Toc385133)

[• head、tail 命令 21](#_Toc385134)

[• grep工具 21](#_Toc385135)

[9．别名 21](#_Toc385136)

[别名：简化复杂的命令 21](#_Toc385137)

[别名案例： 21](#_Toc385138)

[为真机设置永久的别名 22](#_Toc385139)

[ADMIN04 22](#_Toc385140)

[1． wget下载工具软件包 22](#_Toc385141)

[2． RPM Package Manager,RPM包管理器 22](#_Toc385142)

[3．升级内核 23](#_Toc385143)

[下载新的内核软件包 23](#_Toc385144)

[安装新的内核软件包 23](#_Toc385145)

[4．常见安装软件包提示： 23](#_Toc385146)

[软件包已经安装 23](#_Toc385147)

[软件包的依赖关系 23](#_Toc385148)

[5．yum软件仓库 23](#_Toc385149)

[yum配置文件 24](#_Toc385150)

[利用Yum安装软件包 24](#_Toc385151)

[利用Yum卸载软件包 24](#_Toc385152)

[清除Yum缓存 24](#_Toc385153)

[6．配置网络 24](#_Toc385154)

[配置永久的主机名 /etc/hostname 24](#_Toc385155)

[配置永久的IP、子网掩码、网关 25](#_Toc385156)

[7．ssh远程登录 26](#_Toc385157)

[ADMIN05 26](#_Toc385158)

[重定向操作 26](#_Toc385159)

[计算器： bc 26](#_Toc385160)

[查看日期时间：date 26](#_Toc385161)

[管道操作 |： 27](#_Toc385162)

[管理用户 27](#_Toc385163)

[useradd创建用户 27](#_Toc385164)

[id查看用户基本信息 27](#_Toc385165)

[创建用户指定用户id、家目录、登陆解释器、附加组 28](#_Toc385166)

[使用 passwd 命令设置用户登录密码 28](#_Toc385167)

[echo + | 非交互式设置密码 29](#_Toc385168)

[usermode命令修改用户属性 29](#_Toc385169)

[删除用户 29](#_Toc385170)

[管理组 29](#_Toc385171)

[groupadd 命令添加组 29](#_Toc385172)

[gpasswd 命令管理组成员 30](#_Toc385173)

[groupmod 命令修改组属性 30](#_Toc385174)

[groupdel 命令删除组 30](#_Toc385175)

[tar备份与恢复 30](#_Toc385176)

[tar 集成备份工具 30](#_Toc385177)

[制作压缩包,tar命令格式: 31](#_Toc385178)

[释放压缩包,tar命令格式: 31](#_Toc385179)

[显示tar包数据内容 31](#_Toc385180)

[案例2:创建一个备份包 31](#_Toc385181)

[NTP时间同步 31](#_Toc385182)

[1.安装chrony软件与NTP服务器沟通 32](#_Toc385183)

[2.修改主配置文件,指定NTP服务器位置 32](#_Toc385184)

[3.重起程序(服务) 32](#_Toc385185)

[4.验证时间同步 32](#_Toc385186)

[ADMIN06 32](#_Toc385187)

[文档权限--r、w、x 32](#_Toc385188)

[基本权限的类别 32](#_Toc385189)

[权限适用对象(归属) 33](#_Toc385190)

[ls -l 命令查看权限 33](#_Toc385191)

[chmod 命令设置基本权限 33](#_Toc385192)

[Linux判断用户具备的权限: 33](#_Toc385193)

[chmod练习： 34](#_Toc385194)

[文档归属--属主、属组 34](#_Toc385195)

[chown 命令设置文档归属 34](#_Toc385196)

[chown练习 34](#_Toc385197)

[Set GID附加权限(目录的特殊权限) 35](#_Toc385198)

[练习:配置附加权限 35](#_Toc385199)

[Set UID附加权限（适用于可执行文件） 36](#_Toc385200)

[acl访问控制列表 36](#_Toc385201)

[acl访问策略 36](#_Toc385202)

[getfacl、setfacl 命令 37](#_Toc385203)

[ACL设置黑名单 37](#_Toc385204)

[LDAP认证 38](#_Toc385205)

[什么是LDAP? 38](#_Toc385206)

[1.安装软件sssd, 38](#_Toc385207)

[2.图形工具authconfig-gtk进行配置 38](#_Toc385208)

[3.重起sssd服务 38](#_Toc385209)

[4.验证: 39](#_Toc385210)

[家目录漫游 39](#_Toc385211)

[什么是NFS共享 39](#_Toc385212)

[查看NFS资源 39](#_Toc385213)

[ADMIN07 39](#_Toc385214)

[cron计划任务 39](#_Toc385215)

[grep查找文本内容 40](#_Toc385216)

[• 常用命令选项 40](#_Toc385217)

[• 常用的匹配模式 41](#_Toc385218)

[find查找文件 41](#_Toc385219)

[案例 41](#_Toc385220)

[-name "文档名称" 42](#_Toc385221)

[-size +|- 文件大小(k、M、G) 42](#_Toc385222)

[-user 根据用户名 42](#_Toc385223)

[-iname根据名称查找,忽略大小写 42](#_Toc385224)

[-group根据所属组 42](#_Toc385225)

[-mtime根据文件修改时间 42](#_Toc385226)

[• 使用find命令的 -exec 操作 43](#_Toc385227)

# 开学第一天

## 开学典礼、云计算介绍、

讲师：牛犇

第一阶段课程内容共19天

云计算系统管理(Admin)、 云计算应用管理、系统&服务管理进阶

### ADMIN云计算系统管理:

DAY01 开学典礼、云计算介绍、TCP/IP协议及配置

DAY02 Linux系统简介、安装RHEL7系统、RHEL7基本操作

DAY03 命令行基础、 目录和文件管理、 教学环境介绍

DAY04 软件包管理、配置网络、文本/文件查找

DAY05 管理用户和组、tar备份与恢复、NTP时间同步、cron计划任务

DAY06 权限和归属、使用LDAP认证、家目录漫游

DAY07 综合串讲 、综合练习

### ENGINEER云计算应用管理

DAY01 分区规划及使用、LVM逻辑卷

DAY02 Shell脚本基础、使用变量、条件测试及选择、列表式循环

DAY03 系统安全保护、配置用户环境、配置高级连接、防火墙策略管理

DAY04 配置SMB共享、配置NFS共享

DAY05 ISCSI共享存储、数据库服务基础、管理表数据

DAY06 HTTP服务基础、网页内容访问、部署动态网站

DAY07 综合串讲、综合练习

### SERVICES系统&服务管理进阶

DAY01 扩展应用、发布网络YUM源、vim编辑技巧、源码编译安装、systemctl控制

DAY02 DNS服务基础、特殊解析、DNS子域授权、缓存DNS

DAY03 Split分离解析、RAID磁盘阵列、进程管理、日志管理

DAY04 批量装机环境、配置PXE引导、kickstart自动应答

DAY05 rsync同步操作、inotify实时同步、Cobbler网络装机

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

## 教学环境

真机：Linux操作系统 用户名：root 密码：Taren1

教学环境介绍、预装虚拟机说明：

• 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dhcp服务/网关/DNS/软件素材等资源

保证classroom虚拟机优先开机

### 真机使用rht-vmctl辅助工具

• 控制教学用虚拟机

– 格式:rht-vmctl 控制指令 虚拟机名

– 常用控制指令: reset(还原)

]# rht-vmctl reset classroom //先重置资源服务器

]# rht-vmctl reset server

]# rht-vmctl reset desktop //再重置答题虚拟机

虚拟机server与虚拟机desktop的root 密码为redhat

### 虚拟机server：

1.查看系统版本：RHEL7.0

2.查看系统的主机名：server0.example.com

3.查看系统的eth0的IP地址：172.25.0.11/24

虚拟机desktop：

1.查看系统版本：RHEL7.0

2.查看系统的主机名：desktop0.example.com

3.查看系统的eth0的IP地址：172.25.0.10/24

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

真机进行远程管理虚拟机

1.测试真机与虚拟机是否可以通信

[root@room9pc01 ~]# ping 172.25.0.11

[root@room9pc01 ~]# ping 172.25.0.10

2.真机执行ssh进行远程管理虚拟机

格式： ssh root@对方的IP地址

[root@room9pc01 ~]# ssh root@172.25.0.11

[root@room9pc01 ~]# ssh root@172.25.0.10

补充快捷键： Ctrl + shift + t 新开一个终端

3. -X(大写)：在远程管理时开启对方的图形程序

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

[root@server0 ~]# firefox

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

### 虚拟机软件：

虚拟机：利用一款软件，虚拟众多的硬件

win2008虚拟机:密码：Taren1

• C/S,Client/Server架构

– 由服务器提供资源或某种功能

– 客户机使用资源或功能

# 系统管理员ADMIN01

**TCP/IP协议及配置**

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)**

IP地址：唯一标识网络中的一台主机

二进制：由0和1组成

IP地址：32个二进制组成，用点分隔4部分，最后利用4个十进制标示

11111111 00000000 11110000 11110011

192.168.1.1

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

IP地址的概述

**IP地址的分类**

• 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：网络位 + 主机位

192.168.1.1----->C类地址------>网+网+网+主

网络位 :标识一个网络（网段）

192.168.1.1=在192.168.1网络中，有一个编号为1的主机

192.168.1.2=在192.168.1网络中，有一个编号为2的主机

192.168.2.10=在192.168.2网络中，有一个编号为10的主机

子网掩码：用来为计算机标示该IP地址，网络位与主机位

用二进制的0标示主机位，用二进制的1标示网络位

192.168.1.1=11000000.10101000.00000001.00000001

子网掩码: 11111111.11111111.11111111.00000000

IP地址标示： 192.168.1.1 255.255.255.0 = 192.168.1.1/24

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

ifcfg-eth0配置如下：

vim /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0 -----------网卡设备名

HWADDR=00:0C:29:01:4D:22 ---------------MAC地址

TYPE=Ethernet ---------类型为以太网

UUID=39b9e1b8-73b2-4eb3-bb79-72cdbacdd997 ------------唯一识别码

ONBOOT=yes --------------------是否启动网络服务，eth0生效

NM\_CONTROLLED=yes ---------是否可以由network manager图形管理工具托管

BOOTPROTO=static ----------------------是否自动获取ip(none、static、dhcp)

IPADDR=192.168.0.118 ---------------------具体ip地址

NETMASK=255.255.255.0 --------------------子网掩码设置

GATEWAY=192.168.0.1 ----------------------网关

DNS1=111.11.1.1 ---------------------DNS

IPV6INIT=no ---------------------IPV6未开启

USERCTL=no ----------------------禁止非root用户控制次网卡

centos6重启方法

service network restart/stop/start------//从新启动网络配置。

# 系统管理员ADMIN02

## 一：Linux系统简介

Linux是一种操作系统!!

操作系统: 一堆软件的集合，让计算机硬件正常的工作

UNIX诞生,1970-1-1 创始人：

Linux的诞生

• Linux之父,Linus Torwalds 林纳斯.托瓦斯

– 标准读音:“哩呐科斯”

内核 ：调配所有硬件内核，是一个操作系统的核心。是基于硬件的第一层软件扩充， 提供操作系统的最基本的功能，是操作系统工作的基础，它负责管理系统的进 程、内存、设备驱动程序、文件和网络系统，决定着系统的性能和稳定性。

用户------->内核------>调配所有硬件

– 版本号:主版本.次版本.修订号

• 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

• 是一套公开发布的完整Linux系统

– Linux内核 + 各种外围软件（应用软件）

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

1.1 问题

本例要求安装一台可用的KVM服务器：

1. RHEL与CentOS系统有什么关联？

2. RHEL7的根分区默认采用什么文件系统？

1.2 步骤

实现此案例需要按照如下步骤进行。

### 一：RHEL系统与CentOS系统的渊源

RHEL是红帽公司提供的商业版Linux系统，若要获取DVD镜像、安全更新等技术服务支持，一般需要付费订阅；不过构成RHEL系统的各种软件包都是基于GPL开源协议免费发布的。

CentOS是一个社区性质的Linux系统，相当于RHEL的一个克隆版本，它采用了构成RHEL系统的各种软件包重新组装、开发而成，并且在此过程中做了一些优化、必要的Bug修复；CentOS系统的版本会稍晚于同版本的RHEL系统发布，其构成、管理方式与同版本的RHEL系统几乎一模一样，而且能够找到大量开放的软件源，因此受到很多企业的欢迎。

目前，CentOS已经被Red Hat公司所收购，仍然可自由使用。

### 二：RHEL7的根分区默认文件系统

格式化：赋予空间文件系统过程

文件系统（存储数据的规则）

在RHEL7系统中，默认采用XFS文件系统类型；在RHEL6系统中，默认采用EXT4文件系统类型。swap(交换文件系统)虚拟内存：缓解内存不足

### 三：Linux分区表示方式

/dev:存放设备相关数据（硬盘 分区 键盘 鼠标 显示器......）

hd,表示IDE设备

sd,表示SCSI设备

/dev/hda /dev/hdb /dev/hdc /dev/hdd

/dev/sda /dev/sdb /dev/sdc /dev/sdd

/dev/sda1:表示SCSI设备,第一块，第一个分区

/dev/sda5:表示SCSI设备,第一块，第五个分区

/dev/hdc3:表示IDE设备,第三块，第三个分区

## 二：Linux命令行基本操作

本例要求熟悉新装RHEL7系统中命令行界面的获取方法，并通过命令行完成下列任务：

* + 查看内核版本、主机名、IP/MAC地址
  + 查看CPU型号/频率、内存大小
  + 切换到根目录，确认当前位置、列出有哪些子目录
  + 返回到 /root 目录，确认当前位置
  + 重启当前系统

### 一、获取命令行界面的不同方法

#### 虚拟控制台切换

RHEL7系统默认提供6个虚拟控制台（tty1~tty6），每个控制台可以独立登录、执行不同的任务。其中tty1、tty2默认开启图形桌面，tty3~tty6只开启字符模式。

通过组合快捷键Ctrl+Alt+Fn可以在不同的虚拟控制台之间切换，这里的Fn代表F1~F6键中的某一个。例如，当处在正常的图形桌面时，按Ctrl+Alt+F3组合键可以切换到控制台tty3，登录后即进入纯字符模式的命令行界面；如果需要返回之前的图形桌面，则再按键Ctrl+Alt+F1组合键即可。

#### 桌面右键菜单

在桌面空白处右击，或者通过资源管理器浏览文件夹时在空白处右击，可以看到右键菜单中出现“在终端中打开”项（如图-38所示）。

单击“在终端中打开”项即可获得以图形窗口展现的命令行终端程序（如图-39所示）。

#### “应用程序”相应菜单

通过桌面菜单组“应用程序”-->“工具”-->“终端”，也可以打开以图形窗口展现的命令行终端程序（如图-39所示）。

### 二、简单命令行操作练习

#### 查看系统版本：

[root@svr7 桌面]# cat /etc/redhat-release

Red Hat Enterprise Linux Server release 7.2 (Maipo)

#### 列出内核版本：

[root@svr7 桌面]# uname -r

3.10.0-327.el7.x86\_64

#### 列出CPU处理器信息：

[root@svr7 桌面]# lscpu

Architecture: x86\_64

CPU op-mode(s): 32-bit, 64-bit

Byte Order: Little Endian

CPU(s): 1

On-line CPU(s) list: 0

Thread(s) per core: 1

Core(s) per socket: 1

座： 1

NUMA 节点： 1

厂商 ID： GenuineIntel

CPU 系列： 6

型号： 13

型号名称： QEMU Virtual CPU version (cpu64-rhel6)

步进： 3

CPU MHz： 2693.762

BogoMIPS： 5387.52

超管理器厂商： KVM

虚拟化类型： 完全

L1d 缓存： 32K

L1i 缓存： 32K

L2 缓存： 4096K

NUMA 节点0 CPU： 0

#### 检查内存大小、空闲情况

[root@svr7 桌面]# cat /proc/meminfo

MemTotal: 1016904 kB

MemFree: 245364 kB

MemAvailable: 566664 kB

Buffers: 2116 kB

Cached: 417372 kB

SwapCached: 0 kB

Active: 267272 kB

Inactive: 381760 kB

.. ..

#### 切换目录、确认当前位置：

[root@svr7 桌面]# cd /

[root@svr7 /]# pwd

/

#### 图形命令行快捷键：

Ctrl shift + ： 放大 Ctrl - ： 放小

#### 命令行提示标识的含义

– [当前用户@主机名 当前所在位置]$

– 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]$

Lin• 列出网络信息

[root@A /]# ifconfig

[root@A /]# ifconfig eth0 192.168.4.1 #临时设置IP地址

[root@A /]# ifconfig eth0

[root@A /]# ping 192.168.4.1

Ctrl + c ：结束正在运行的指令

127.0.0.1:永远代表本机

## 三：Linux 的五个重启命令

　　1、shutdown

　　2、poweroff

　　3、init

　　4、reboot

5、halt

　　在linux下一些常用的关机/重启命令有shutdown、halt、reboot、及init，它们都可以达到重启系统的目的，但每个命令的内部工作过程是不同的。

### 1.shutdown

　　shutdown命令安全地将系统关机。 有些用户会使用直接断掉电源的方式来关闭linux，这是十分危险的。因为linux与windows不同，其后台运行着许多进程，所以强制关机可能会导致进程的数据丢失﹐使系统处于不稳定的状态﹐甚至在有的系统中会损坏硬件设备。而在系统关机前使用shutdown命令﹐系统管理员会通知所有登录的用户系统将要关闭。并且login指令会被冻结﹐即新的用户不能再登录。直接关机或者延迟一定的时间才关机都是可能的﹐还可能重启。这是由所有进程〔process〕都会收到系统所送达的信号〔signal〕决定的。这让像vi之类的程序有时间储存目前正在编辑的文档﹐而像处理邮件〔mail〕和新闻〔news〕的程序则可以正常地离开等等。

　　shutdown执行它的工作是送信号〔signal〕给init程序﹐要求它改变runlevel。

　　Runlevel 0被用来停机〔halt〕﹐runlevel 6是用来重新激活〔reboot〕系统﹐而runlevel 1则是被用来让系统进入管理工作可以进行的状态﹔这是预设的﹐假定没有-h也没有-r参数给shutdown。要想了解在停机〔halt〕或者重新开机〔reboot〕过程中做了哪些动作﹐你可以在这个文件/etc/inittab里看到这些runlevels相关的资料。

shutdown 参数说明:

[-t] 在改变到其它runlevel之前﹐告诉init多久以后关机。

[-r] 重启计算器。

[-k] 并不真正关机﹐只是送警告信号给每位登录者〔login〕。

[-h] 关机后关闭电源〔halt〕。

[-n] 不用init﹐而是自己来关机。不鼓励使用这个选项﹐而且该选项所产生的后果往往不总是你所预期得到的。

[-c] cancel current process取消目前正在执行的关机程序。所以这个选项当然没有时间参数﹐但是可以输入一个用来解释的讯息﹐而这信息将会送到每位使用者。

[-f] 在重启计算器〔reboot〕时忽略fsck。

[-F] 在重启计算器〔reboot〕时强迫fsck。

[-time] 设定关机〔shutdown〕前的时间。

shutdown -r now 立刻重启([root](https://www.baidu.com/s?wd=root&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)用户使用)

shutdown -r 10 过10分钟自动重启(root用户使用)

shutdown -r 20:35 在时间为20:35时候重启(root用户使用)

shutdown -h now 立刻关机(root用户使用)

shutdown -h 10 10分钟后自动关机

shutdown –p 2 两分钟后关闭电源

### 2.halt----最简单的关机命令

　　其实halt就是调用shutdown -h。halt执行时﹐杀死应用进程﹐执行sync系统调用﹐文件系统写操作完成后就会停止内核。

　　参数说明:

[-n] 防止sync系统调用﹐它用在用fsck修补根分区之后﹐以阻止内核用老版本的超级块〔superblock〕覆盖修补过的超级块。

[-w] 并不是真正的重启或关机﹐只是写wtmp〔/var/log/wtmp〕纪录。

[-d] 不写wtmp纪录〔已包含在选项[-n]中〕。

[-f] 没有调用shutdown而强制关机或重启。

[-i] 关机〔或重启〕前﹐关掉所有的网络接口。

[-p] 该选项为缺省选项。就是关机时调用poweroff。

### 3.reboot

　　reboot的工作过程差不多跟halt一样﹐不过它是引发主机重启﹐而halt是关机。它 的参数与halt相差不多。

### 4.init

语法：init(选项)(参数)

**init选项：**

-b：不执行相关脚本而直接进入单用户模式；

-s：切换到单用户模式。

**init参数：**

0 停机（千万不能把initdefault 设置为0）

1 单用户模式

2 多用户，没有 NFS(和级别3相似，会停止部分服务)

3 完全多用户模式

4 没有用到

5 x11(Xwindow)

1. 重新启动（千万不要把initdefault 设置为6）

### 5.poweroff---立即关机

## 四：IP地址的概述

### • IP地址的分类

• 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

10.11.12.13-------》IP地址的第一个数字10-------》A类

192.168.1.1-------》IP地址的第一个数字192-------》C类

IP地址：网络位 + 主机位

192.168.1.1----->C类地址------>网+网+网+主

网络位 :标识一个网络（网段）

192.168.1.1=在192.168.1网络中，有一个编号为1的主机

192.168.1.2=在192.168.1网络中，有一个编号为2的主机

192.168.2.10=在192.168.2网络中，有一个编号为10的主机

子网掩码：用来为计算机标示该IP地址，网络位与主机位

用二进制的0标示主机位，用二进制的1标示网络位

192.168.1.1=11000000.10101000.00000001.00000001

子网掩码: 11111111.11111111.11111111.00000000

IP地址标示： 192.168.1.1 255.255.255.0 = 192.168.1.1/24

网关地址作用： 从一个网络连接到另一个网络的“关口”解决不同网络之间通信

# 系统管理员ADMIN03

案例1：命令行基础技巧

案例2：挂载并访问光盘设备

案例3：ls列表及文档创建

案例4：复制、删除、移动

## • Linux命令

– 用来实现某一类功能的指令或程序

Linux执行命令，需要找到命令的执行程序

– 命令的执行依赖于解释器(默认的解释器：/bin/bash)

用户操作-------》解释器-------》内核-------》调配所有硬件

执行顺序:

别名 函数 内部 外部

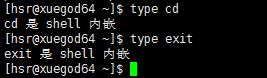
**• Linux命令的分类**

Linux命令有内部命令（内建命令）和外部命令之分，内部命令和外部命令功能基本相同，但也有些细微差别。

**– 内部命令:属于解释器的一部分**

内部命令不需要使用子进程来执行，它们已经和shell编译成一体，作为shell工具的组成部分存在。不需要借助外部程序文件来运行。它们是一些比较简单的linux系统命令，如exit,history,cd,echo等。

　　要分区是外部命令还是内部命令可以使用type 来查看,如下图查看cd是内部还是外部命令：



[root@client50 ~]#enable #查看所有内部命令

enable .

enable :

enable [

enable alias

enable bg

enable bind

.....

**– 外部命令:解释器之外的其他程序**

外部命令是存在于bash shell之外的程序，它们不是shell程序的一部分，外部命令程序通常位于/bin, /usr/bin, /sbin, /usr/sbin中。 ps就是一个外部命令，使用type查看如下所示:

180923154953565

　　所有外部命令都会被一个子进程来执行。ps的父进程是bash shell PID 2486，如下所示：

180923154953566

绿色：可以执行的程序 青色：快捷方式 黑色：文件 蓝色：目录

## 1．清理编辑的命令基础技巧

练习以下快捷编辑操作：Ctrl + l、Ctrl + u、Ctrl + w；Ctrl + c、Esc + .

* Esc + . 或 Alt + . : 粘贴上一个命令的参数
* Ctrl + l:清空整个屏幕
* Ctrl + c:废弃当前编辑的命令行(结束正在运行的指令)
* Ctrl + u:清空至行首(以光标位置界定)
* Ctrl + w:往回删除一个单词(以空格界定)
* Ctrl + k:往后删除(已光标位置界定)
* Ctrl + shift + t 新开一个终端

## 2．mount挂载

### mount挂载概念：

mount挂载操作：让目录成为设备的访问点

从安装系统的光盘中获得众多的软件包

Windwos访问光盘内容：

光盘文件------》光驱设备------》CD驱动器（双击）(访问点)

Linux访问光盘内容：

光盘文件------》光驱设备------》访问点（目录）/dev/cdrom

在Linux访问设备内容必须通过访问点

青色：快捷方式

总结：

1.Linux访问设备内容需要通过访问点

2.Linux访问点为一个目录

3.通过mount挂载命令，让目录成为设备的访问点

### mount案例：挂载并访问光盘设备

2.1 问题

本例要求学会mount挂载操作。主要完成下列任务：

连接光盘 /ISO/rhel-server-7.4-x86\_64-dvd.iso

将光盘挂载到 /mnt 目录，检查 /mnt 目录内容

卸载光盘设备，再次检查目录内容

#### 2.1）临时挂载

连接光盘 /ISO/rhel-server-7.4-x86\_64-dvd.iso

[root@server0 ~]# mount /dev/cdrom /mnt //挂载设备

mount: /dev/sr0 写保护，将以只读方式挂载

#### 2.2）挂载后访问

将光盘挂载到 /mnt 目录，检查 /mnt 目录内容

[root@server0 ~]# ls /mnt //访问设备内容

addons images Packages RPM-GPG-KEY-redhat-release

EFI isolinux release-notes TRANS.TBL

EULA LiveOS repodata

#### 2.3）卸载设备

卸载光盘设备，再次检查目录内容

[root@server0 ~]# umount /mnt //卸载设备

mount: /dev/sr0 写保护，将以只读方式挂载

[root@server0 ~]# ls /mnt //确认结果

[root@server0 ~]#

注意事项：卸载时当前路径不要是挂载点目录

[root@A dvd]# umount /dvd

umount: /dvd：目标忙。

(有些情况下通过 lsof(8) 或 fuser(1) 可以找到有关使用该设备的进程的有用信息)

[root@A dvd]#

Linux允许一个设备有多个挂载点，不允许一个目录有多个设备挂载

挂载：建议不要挂载根目录下，已存在的目录，最好自己创建目录

#### 2.4）开机自动挂载

修改/etc/fstab 文件，文件格式如下：

– 设备路径 挂载点 类型 参数 备份标记 检测顺序

[root@server0 ~]# mount –a #fstab文件是否正确

1. 挂载 rhel-server-7.2-x86\_64-dvd.iso 到/var/www/html/rh7dvd

[root@room9pc13 ~]# vim /etc/fstab

.. .

/ISO/rhel-server-7.2-x86\_64-dvd.iso /var/www/html/rh7dvd iso9660 loop,ro 0 0

[root@room9pc13 ~]# mount -a

## 3．ls列表及文档创建

### ls — List格式

– 格式:ls [选项]... [目录或文件名]

• 常用命令选项

– -l:以长格式显示,显示详细信息

– -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)

### 文档开头判断

以 - 开头:文本文件

以 d 开头:目录

以 l 开头:快捷方式

## 4．目录管理

### mkdir 创建目录

• mkdir — Make Directory

– 格式:mkdir [-p] [/路径/]目录名...

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

#选项-p 如果vod、movie目录都没有，连同一起创建

### cd切换目录

cd — Change Directory

– 用途:切换工作目录

– 格式:cd [目标文件夹位置]

. 表示当前目录 .. 表示父目录 ~user表示用户user的家目录

～：代表用户家目录（家目录：存放用户个性化信息的目录）

Linux管理员root家目录： /root

存放所有普通用户家目录： /home

### touch创建文件

touch –C 已有文件名 #可修改文件时间，不可创建文件

### 4.1 目录管理案例

[root@A /]# cd ~root #切换到root用户家目录

[root@A ~]# pwd

[root@A /]# useradd zhangsan #创建用户zhangsan

[root@A /]# cd ~zhangsan #切换到zhangsan用户家目录

[root@A zhangsan]# pwd

## 5．通配符使用

针对不确定的文档名称,以特殊字符表示

\*：任意多个任意字符

?：单个字符

[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

### 5.1 通配符案例

[root@A ~]# ls /boot/vm\* #以vm开头

[root@A ~]# ls /etc/\*tab #以tab结尾

[root@A ~]# ls /etc/\*.conf #以.conf结尾

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

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

[root@A ~]# ls /etc/??tab

[root@A ~]# ls /dev/tty[3-8]

[root@A ~]# ls /dev/tty[1-9]

[root@A ~]# ls /dev/tty{1,5,9,12}

[root@A ~]# ls /dev/tty{21,23,28,29,S0}

[root@A ~]# ls /dev/tty{2[0-9],30} # 显示/dev/tty20至/dev/tty30

[root@A ~]# ls /dev/tty{2?,30}

## 6．Vim编辑器

vim是Linux系统上最常用的命令行交互式文本编辑器，主要工作在三种模式：命令模式、输入模式、末行模式。

通过vim打开一个文件时，默认处于命令模式；从命令模式按i键可以进入编辑状态，按Esc键返回命令模式；从命令模式输入冒号:可以进入末行模式，在末行模式下主要执行存盘、退出等基本操作。

vim文本编辑器

三个模式：命令模式 插入模式（编辑模式,输入模式） 末行模式

具备创建文件功能

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

命---------i------》插入模式(按Esc 回到命令模式)

令

模

式---------：---------》末行模式(按Esc 回到命令模式)

末行模式 ：wq 保存并退出

末行模式 ：q！ 强制不保存并退出

### 步骤一：使用ls命令列出指定的文件

#### 1）列出/etc/目录下以re开头.conf结尾的文件

使用通配符 \* 代替未知的字符串。

[root@server0 ~]# ls /etc/re\*.conf

/etc/request-key.conf /etc/resolv.conf

#### 2）列出/dev/目录下编号是个位数的tty控制台设备

使用通配符 ? 代替单个未知的字符。

[root@server0 ~]# ls /dev/tty?

/dev/tty0 /dev/tty2 /dev/tty4 /dev/tty6 /dev/tty8

/dev/tty1 /dev/tty3 /dev/tty5 /dev/tty7 /dev/tty9

或者更严谨一些，使用 [0-9] 代替单个数字。

[root@server0 ~]# ls /dev/tty[0-9]

/dev/tty0 /dev/tty2 /dev/tty4 /dev/tty6 /dev/tty8

/dev/tty1 /dev/tty3 /dev/tty5 /dev/tty7 /dev/tty9

### 步骤二：mkdir新建文档

#### 1）使用mkdir新建文件夹

[root@server0 ~]# mkdir -p /protected/project/tts10

[root@server0 ~]# ls -ld /protected/project/tts10/

drwxr-xr-x. 2 root root 6 Aug 30 10:11 /protected/project/tts10/

#### 2）使用vim新建或修改文本文件

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

//按i键进入编辑模式

//将文本内容修改为 svr7.tedu.cn

//按Esc键返回命令模式

//输入:wq保存修改并退出vim编辑器

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

svr7.tedu.cn

## 7．复制、删除、移动

### rm 删除

• rm — Remove

– 格式:rm [选项]... 文件或目录...

• 常用命令选项

– -r:递归删除(含目录)、-f:强制删除

#实际中可卸载rm，用时再装，用完再删。

### mv 移动/改名

• mv — Move

– 格式:mv [选项]... 原文件... 目标路径

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

[root@A ~]# ls /opt/

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

重命名：将路径不变的移动

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

[root@A ~]# ls /opt/

[root@A ~]# mv /opt/student/ /opt/test

[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

-p :复制是将文件的权限属性一并复制

[root@A ~]# cp /etc/passwd /opt/

[root@A ~]# ls /opt/

[root@A ~]# cp -r /boot/ /opt/ #复制目录必须要加上-r选项

[root@A ~]# ls /opt/

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

复制出现重名：在本次操作中，临时取消别名

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

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

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

复制支持两个以上的参数：永远把最后一个参数作为目标，其他所有参数作为源

[root@A ~]# cp /etc/passwd /etc/shadow /etc/fstab /etc/hosts /etc/redhat-release /mnt/

[root@A ~]# ls /mnt/

复制与 一个点 . 的连用: 将源文档 复制到 当前路径下

[root@A CA]# cp /etc/hosts .

[root@A CA]# ls

[root@A CA]# cp /etc/fstab . #复制到当前目录

[root@A CA]# ls

复制时可以重新命名，目标路径的新文档名称

[root@A /]# cp -r /home/ /opt/test #将/home复制到/opt重新命名为test目录

[root@A /]# ls /opt/

[root@A /]# cp -r /home/ /opt/test #将/home复制到/opt/test目录下

## 8．查看大文本内容操作

### • less分屏阅读工具

• 格式:less [选项] 文件名...

– 优势:支持前后翻页

• 基本交互操作

– 按上下键进行滚动

– 按 / 键向后查找

– 按 q 键退出

[root@A /]# less /etc/passwd 输入 /root 全文查找root

### • head、tail 命令

– 格式:

head -n 数字 文件名 查看前n行内容

tail -n 数字 文件名 查看后n行内容

[root@A /]# head -3 /etc/passwd

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

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

[root@A /]# tail -3 /etc/passwd

[root@A /]# head /etc/passwd #默认是显示前10行

[root@A /]# tail /etc/passwd #默认是显示后10行

### • grep工具

– 用途:输出包含指定字符串的行

– 格式:grep '查找条件' 目标文件

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

[root@A /]# grep haxi /etc/passwd #没有输出信息

## 9．别名

### 别名：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]

临时取消别名 命令前面加“ \ ”

别名案例：

[root@A ~]# hostname

[root@A ~]# hn

bash: hn: 未找到命令...

[root@A ~]# alias hn='hostname' #定义别名

[root@A ~]# alias #查看当前系统生效的别名

[root@A ~]# unalias hn #删除别名

[root@A ~]# hn

bash: hn: 未找到命令...

[root@A ~]# alias myls='ls -ld'

[root@A ~]# myls /root

### 为真机设置永久的别名

1.修改别名配置文件

[root@room9pc01 ~]# vim /root/.bashrc #配置完成后开启新的终端生效

# .bashrc

alias gos='ssh -X root@172.25.0.11'

alias god='ssh -X root@172.25.0.10'

开启新的终端进行验证

Vim /root/.bashrc #只对root用户有效

Vim /home/student/.bashrc #只对student用户有效

Vim /etc/bashrc #对所有用户有效

在空白处添加内容： alias 自定义名=功能 ，知识后重启终端生效

用户个性化配置文件

• 影响指定用户的 bash 解释环境

– ~/.bashrc,每次开启 bash 终端时生效

全局环境配置(别名)

• 影响所有用户的 bash 解释环境

– /etc/bashrc,每次开启 bash 终端时生效,对所有用户生效的别名

[root@server0 ~]# vim /root/.bashrc

alias hello='echo hello'

[root@server0 ~]# vim /home/student/.bashrc

alias hi='echo hi'

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

alias abc='echo abc'

[root@server0 ~]# hello #执行成功

[root@server0 ~]# hi #执行失败

[root@server0 ~]# abc #执行成功

[root@server0 ~]# su - student

[student@server0 ~]$ hello #执行失败

[student@server0 ~]$ hi #执行成功

[student@server0 ~]$ abc #执行成功

[student@server0 ~]$ exit

# 系统管理员ADMIN04

## wget下载工具软件包

– wget 软件包的URL网址 -O /目标目录/重命名 #默认下载到当前路径下

-O(大写)：下载另存为

## 2． RPM Package Manager,RPM包管理器

– rpm -q 软件名... #查询软件包是否安装

-- rpm -ql 软件名 #查询已安装软件的安装清单

– rpm -ivh 软件名-版本信息.rpm... #安装软件包

– rpm -e 软件名... #卸载程序

– rpm -Uvh 软件名-版本信息.rpm... #升级安装软件包

[root@server0 ~]# rpm -q vsftpd #查询软件是否安装

未安装软件包 vsftpd

[root@server0 ~]# rpm -q firefox

[root@server0 ~]# rpm -q zip

[root@server0 ~]# rpm -ivh vsftpd-3.0.2-9.el7.x86\_64.rpm

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

[root@server0 ~]# rpm -ql vsftpd #查询已安装软件的安装清单

### 导入红帽签名（了解内容）

[root@server0~]#wget \ http://classroom.example.com/content/rhel7.0/x86\_64/dvd/RPM-GPG-KEY-redhat-release

#下载红帽签名

[root@server0 ~]# rpm --import RPM-GPG-KEY-redhat-release #导入红帽签名

[root@server0 ~]# rpm -ivh vsftpd(tab)

## 3．升级内核

### 下载新的内核软件包

[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 ~]# ls

### 安装新的内核软件包

]# uname –r #安装前查看一下现有内核版本

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

]# reboot #重起系统刷新

[root@room9pc01 ~]# gos

[root@server0 ~]# uname –r #安装后再查看内核版本变化

3.10.0-123.1.2.el7.x86\_64

[root@server0 ~]#

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

## 4．常见安装软件包提示：

### 软件包已经安装

[root@server0 ~]# rpm -ivh vsftpd-3.0.2-9.el7.x86\_64.rpm

准备中... ################################# [100%]

软件包 vsftpd-3.0.2-9.el7.x86\_64 已经安装

[root@server0 ~]# LANG=en #修改系统语言环境为英文

[root@server0 ~]# rpm -ivh vsftpd-3.0.2-9.el7.x86\_64.rpm9

Preparing... ################################# [100%]

package vsftpd-3.0.2-9.el7.x86\_64 is already installed

[root@server0 ~]#

### 软件包的依赖关系

]# rpm -ivh bind-chroot-9.9.4-14.el7.x86\_64.rpm #rpm安装无法安装依赖包

错误：依赖检测失败：

bind = 32:9.9.4-14.el7 被 bind-chroot-32:9.9.4-14.el7.x86\_64 需要

]#

## 5．yum软件仓库

作用：自动解决依赖关系安装软件

服务：为客户端自动解决依赖关系安装软件包

服务端：虚拟机classroom

1.众多的软件包 2.仓库数据文件 3.具备光盘所有内容

4.搭建FTP服务或Web服务，共享光盘所有内容

客户端：虚拟机server

书写Yum客户端配置文件，指定服务端位置

### yum配置文件

配置文件要求：/etc/yum.repos.d/\*.repo 在此目录下,错误的文件会影响正确的文件

[root@server0 ~]# mv /etc/yum.repos.d/\*.repo /etc/yum.repos.d/repobak/

[root@server0 ~]# rm -rf /etc/yum.repos.d/\* #删除前备份原有yum配置文件

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

[rhel7] #仓库标识

name=RHEL7.0 #仓库描述信息

baseurl=http://classroom.example.com/content/rhel7.0/x86\_64/dvd/ #指定服务端位置

enabled=1 #启用本文件

gpgcheck=0 #不检测红帽签名（GPG签名检查）

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

### 利用Yum安装软件包

[root@server0 ~]# yum -y install 软件名

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

### 利用Yum卸载软件包

[root@server0 ~]# yum remove httpd

### 清除Yum缓存

[root@server0 ~]# yum clean all

## 6．配置网络

### 配置永久的主机名 /etc/hostname

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

nsd.tedu.cn

[root@server0 ~]# hostname #查看本机主机名

nsd.tedu.cn

[root@server0 ~]# exit #修改后要重新登入才能生效

登出

Connection to 172.25.0.11 closed.

[root@room9pc01 ~]# gos

Last login: Tue Dec 4 15:38:47 2018 from 172.25.0.250

[root@nsd ~]#

### 配置永久的IP、子网掩码、网关

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

#### 1.直接修改ifcfg-eth0配置文件

#### 2.利用命令方式配置

**a：查看命令识别的网卡名**

]# nmcli connection show #查看网卡设备

[root@server0 ~]# nmcli connection show

名称 UUID 类型 设备

System eth0 5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03 802-3-ethernet eth0

**b:配置IP地址、子网掩码、网关地址**

]# 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 '网卡名' ipv4.方法 手工配置

ipv4.地址 ' IP地址/子网掩码 网关地址' 每次开机自动启用本网卡所有配置

**c:激活配置**

]# nmcli connection up 'System eth0'

#### 3.检查修改结果，确认无误

**检查IP地址：**

[root@server0 ~]# ifconfig eth0

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

inet 172.25.0.11 netmask 255.255.255.0 broadcast 172.25.0.255

inet6 fe80::5054:ff:fe00:b prefixlen 64 scopeid 0x20<link>

ether 52:54:00:00:00:0b txqueuelen 1000 (Ethernet)

RX packets 1394 bytes 138855 (135.6 KiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 944 bytes 98495 (96.1 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

**检查默认网关地址：**

[root@server0 ~]# route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

0.0.0.0 172.25.0.254 0.0.0.0 UG 1024 0 0 eth0

172.25.0.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

### 配置DNS服务器地址

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

# Generated by NetworkManager

search example.com

nameserver 172.25.254.254

如果在使用nmcli修改网络连接时并未指定ipv4.dns，也可以直接修改DNS客户端配置文件/etc/resolv.conf，确保添加有上述记录即可。

[root@server0 ~]# nslookup www.qq.com #测试DNS解析命令

### 修改配置文件:

vim /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0 -----------网卡设备名

HWADDR=00:0C:29:01:4D:22 ---------------MAC地址

TYPE=Ethernet ---------类型为以太网

UUID=39b9e1b8-73b2-4eb3-bb79-72cdbacdd997 ------------唯一识别码

ONBOOT=yes --------------------是否启动网络服务，eth0生效

NM\_CONTROLLED=yes ---------是否可以由network manager图形管理工具托管

BOOTPROTO=static ----------------------是否自动获取ip(none、static、dhcp)

IPADDR=192.168.0.118 ---------------------具体ip地址

NETMASK=255.255.255.0 --------------------子网掩码设置

GATEWAY=192.168.0.1 ----------------------网关

DNS1=111.11.1.1 ---------------------DNS

IPV6INIT=no --------------------IPV6未开启

USERCTL=no ---------------------禁止非root用户控制次网卡

修改成

DEVICE="eth0"

ONBOOT="yes"

TYPE="Ethernet"

BOOTPROTO="static"

IPADDR="192.168.1.10"

NETMASK="255.255.255.0"

GATEWAY="192.168.1.254"

## 7．ssh远程登录以及免密登录

重置过的练习环境已预先配置好网络，并且为从真机访问答题用虚拟机提前配置了SSH密钥验证，因此直接执行快速登录（ssh -X root@目标主机地址）即可。

[root@room9pc13 ~]# ssh -X root@server0.example.com 远程主机上登陆到server0

[root@server0 ~]# hostname

server0.example.com

[root@room9pc13 ~]# ssh -X root@172.25.0.11 远程主机上登陆到172.25.0.11

[root@server0 ~]# hostname

server0.example.com

注意ssh添加了-X选项（大写字母X），这是为了在执行远程主机的图形程序时，能够将图形界面在客户机上显示，方便用户操作。

例如，连接到server0以后，运行对方的网卡配置工具nm-connection-editor、firefox等其程序窗口会直接显示在客户机的图形桌面上。

**免密登录**

先生成本地秘钥,在发送秘钥给远程主机

Ssh-keygen生成秘钥 发送秘钥给远程主机ssh-copy-id

[root@sql09 ~]# ssh-keygen #一路回车即可

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id\_rsa): #直接回车

Created directory '/root/.ssh'.

Enter passphrase (empty for no passphrase): #直接回车

Enter same passphrase again:

Your identification has been saved in /root/.ssh/id\_rsa.

Your public key has been saved in /root/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:Ki15jYWZsvqrkXrt+912gmvbxIJgb5YAomZJZbCakKs root@sql09

The key's randomart image is:

+---[RSA 2048]----+

| ..o |

| .+ |

|+o. |

|=+.. + |

|=+ = + S |

|+ o O B . |

|E o.= @ o.o |

| ..o.\* .o=o . |

|..o+=+..++oo |

+----[SHA256]-----+

[root@sql09 ~]# ssh-copy-id 192.168.4.60

Are you sure you want to continue connecting (yes/no)? Yes #输入yes

# 系统管理员ADMIN05

## 重定向操作

作用:将前面命令的输出，当作内容写入到文本文件中

> 覆盖重定向

>> 追加重定向

]#ls –help > /opt/ls.txt #将ls的帮助内容重定向到ls.txt中

## 计算器： bc

常见运算： 加+ 减- 乘\* 除 / 取余运算 %

]#bc 进入运算界面

## 查看日期时间：date

[root@A ~]# date

[root@A ~]# date +%Y

[root@A ~]# date +%m

[root@A ~]# date +%d

[root@A ~]# date +%Y%m%d

[root@A ~]# date +%Y-%m-%d

[root@A ~]# date +%F

修改时间：date –s '年-月-日 时:分:秒'

[root@A ~]# date -s '2008-10-1 11:11:11'

## 管道操作 |：

作用：将前面命令的输出结果，传递到后面命令，作为后面命令的参数

显示文件8到12行/etc/passwd

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

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

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

[root@A ~]# cat -n /etc/passwd | head -12 | tail -5 #前面12行的最后5行即为：8-12行

[root@A ~]# echo 1+1 | bc #输出的结果1+1管道给bc工具计算

[root@A ~]# ls --help | less #ls命令的帮助信息分屏查看

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

[root@A ~]# ifconfig | grep eth0 #查看网卡信息，并只筛选出eth0的网卡

xargs 可以将前一命令的输出转换成参数

ls 不接受输入,ls 只接受参数,

]# ls /etc/hosts >/tmp/h.txt

]# cat /tmp/h.txt

/etc/hosts

]# cat /tmp/h.txt | ls -l #查看的是当前所在目录的所有文件的权限

]#cat /tmp/h.txt | xargs ls -l #查看的是/etc/hosts的权限

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

## 管理用户

用户唯一标识：UID 管理员root的UID： 0

组唯一标识 ：GID 组：方便对用户的管理，方便对权限管理

Linux的用户至少属于一个组

基本组（主组）：创建用户默认新增，组名与用户名相同

附加组（从属组）：一个用户可加入多个附加组

### useradd 模板介绍

在linux系统中添加新用户的工具是useradd, 可以一次性创建新用户账户及设置用户HOME目录结构。useradd命令使用系统的默认值以及命令行参数来设置用户账户。系统默认值被设置在/etc/default/useradd文件中，可以定位到/etc/default/useradd文件,修改默认值。

[zdd1@rsync ~]$ cat /etc/default/useradd

# useradd defaults file

GROUP=100

HOME=/home

INACTIVE=-1

EXPIRE=

SHELL=/bin/bash

SKEL=/etc/skel

CREATE\_MAIL\_SPOOL=yes

使用useradd -D 选项查看这些默认值。

[root@rsync ~]# useradd -D

GROUP=100

HOME=/home

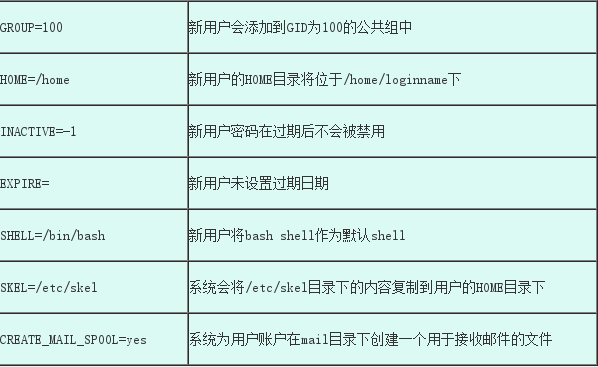
INACTIVE=-1

EXPIRE=

SHELL=/bin/bash

SKEL=/etc/skel

CREATE\_MAIL\_SPOOL=yes



SKEL参数是指：useradd命令允许管理员创建一份默认的HOME目录配置，然后把它作为创建新用户HOME目录的模板。这样就能自动在每个新用户的HOME目录里放置默认的系统文件(上篇讲到的系统环境变量位置)。

[root@rsync ~]# ls -al /etc/skel/

总用量 24

drwxr-xr-x. 3 root root 91 3月 27 19:30 .

drwxr-xr-x. 139 root root 8192 3月 27 19:30 ..

-rw-r--r--. 1 root root 0 3月 27 19:30 1.txt

-rw-r--r--. 1 root root 18 3月 8 2017 .bash\_logout

-rw-r--r--. 1 root root 193 3月 8 2017 .bash\_profile

-rw-r--r--. 1 root root 231 3月 8 2017 .bashrc

drwxr-xr-x. 4 root root 39 1月 30 2018 .mozilla

.bashrc, .bash\_profile这些文件都是bash shell的标准启动文件。系统会自动将这些默认文件复制到你创建的每个用户HOME目录下。

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

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值:用户描述信息:用户的家目录:解释器

• 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 登录解释器程序、

-G 附加组

-c 给新用户添加备注

-e 用YYYY-MM-DD格式指定一个账户过期日期

-f 指定这个账户密码过期后多少天，这个账户被禁用;0表示密码一过期就立即禁用, 1表示禁用这个功能

-r 创建系统账户

-p 指定默认密码

当创建用户账户时，如果总需要修改某个值的话，最好还是修改一下系统的默认值，修改(useradd模板),也可以使用"useradd -D 默认参数"来修改模板，这里就不再介绍useradd默认值参数。

### useradd创建用户

[root@A ~]# useradd nsd02

### id查看用户基本信息

[root@A ~]# id nsd02 #id命令查看

[root@A ~]# grep nsd02 /etc/passwd #直接查看文件

### 创建用户指定用户id、家目录、登陆解释器、附加组

[root@A ~]# useradd -u 1500 nsd05

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

[root@A ~]# useradd -d /mnt/nsd07 nsd07

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

[root@A ~]# ls /mnt/

用户名:密码占位符:UID:基本组的GID值:用户描述信息:用户的家目录:解释器

[root@A ~]# groupadd tarena

[root@A ~]# useradd -G tarena nsd08

[root@A ~]# id nsd08

/sbin/nologin:禁止用户登陆系统

[root@A ~]# useradd -s /sbin/nologin nsd09

[root@A ~]# grep nsd09 /etc/passwd

### 使用 passwd 命令设置用户登录密码

passwd 参数 用户名

-l：锁定已经命名的账户名称

-u：解开账户锁定状态

-x, --maximum=DAYS：密码使用最大时间（天）

-n, --minimum=DAYS：密码使用最小时间（天）

-d：删除使用者的密码

-S：检查指定使用者的密码认证种类

--stdin：非交互式修改/设置密码，弊端是操作日志能查密码，用history -c 干掉。

用户密码信息存放在 /etc/shadow 文件

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

root:$6$UiGI4Tc2$htsXYn5cJnOqv3P1VLcUSgfjDu2pL5yiJBuua6foZAHdwqeuLHfYUfS/vBn27Wjvoel8EJgtdsMjyquqvKAmf1:16261:0:99999:7:::

1用户名:2 密码加密字符串:3上一次修改密码的时间：4密码最短使用天数：5密码使用最长天数：6提前警告密码过期天数：7:8:9

– passwd [用户名]

在命令行临时切换用户身份: su - 用户名

Root可以重置普通用户的密码，且不需要验证旧密码，可忽略密码复杂度要求

普通用户修改密码需要验证旧密码

[root@A ~]# passwd nsd01

更改用户 nsd01 的密码 。

新的 密码： #输入新密码

重新输入新的 密码： #重新输入新密码

passwd：所有的身份验证令牌已经成功更新。

[root@A ~]# su - 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: #重新输入新密码

passwd: all authentication tokens updated successfully.

[nsd01@A ~]$ exit

### echo + | 非交互式设置密码

– echo '密码' | passwd --stdin 用户名

--stdin 这个选项用于 从标准输入 管道读入新的密码

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

### usermode命令修改用户属性

• 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

-c：修改用户帐号的备注文字。

-e：修改帐号的有效期限。

-f： 改在密码过期后多少天即关闭该帐号。

-g：修改用户所属的群组。

-l：修改用户帐号名称。

-L：锁定用户密码，使密码无效。

-U：解除密码锁定。

[root@A ~]# useradd nsd11

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

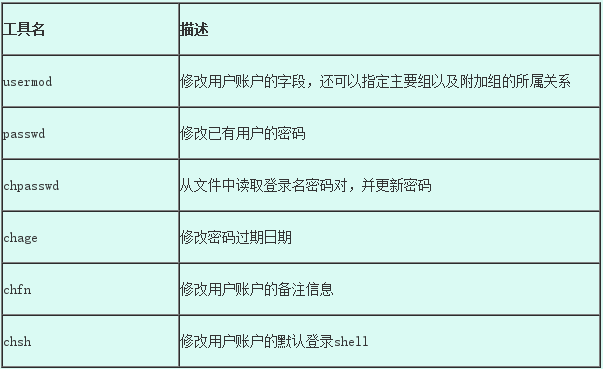
nsd11:x:1505:1506::/home/nsd11:/bin/bash

]# usermod -u 1600 -d /opt/nsd11 -s /sbin/nologin -G tarena nsd11

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

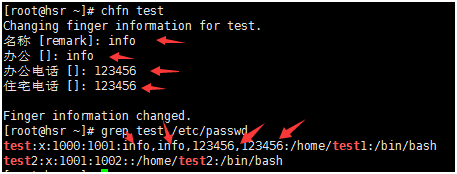
nsd11:x:1600:1506::/opt/nsd11:/sbin/nologin

[root@A ~]# id nsd11



(1) chpasswd ：如果要大量用户修改密码，该工具命令可以事半功倍。

(2) chsh : 该命令用来快速修改默认的用户登录shell。例如 chsh -s /bin/csh test 。

(3) chfn：该命令会向你询问要将哪些适合的内容加入备注字段，下面给test用户加内容备注，如图所示：

### chage命令修改账户有效期

该命令用来管理用户账户的有效期，你需要对每个值设置多个参数，列表如下：



[root@rsync ~]# chage zdd -E 2019-10-10

[root@rsync ~]# chage -E 2019-10-10 zdd1

[root@rsync ~]# chage -l zdd

最近一次密码修改时间 ：3月 27, 2019

密码过期时间 ：从不

密码失效时间 ：从不

帐户过期时间 ：10月 10, 2019

两次改变密码之间相距的最小天数 ：0

两次改变密码之间相距的最大天数 ：99999

在密码过期之前警告的天数 ：7

[root@rsync ~]# chage -l zdd1

最近一次密码修改时间 ：3月 27, 2019

密码过期时间 ：从不

密码失效时间 ：从不

帐户过期时间 ：10月 10, 2019

两次改变密码之间相距的最小天数 ：0

两次改变密码之间相距的最大天数 ：99999

在密码过期之前警告的天数 ：7

### 用户密码立即失效

chage -d0 tom #用户密码立即失效

### 生成随机字符串(作为密码)

openssl rand -base64 10

### userdel删除用户

• 使用 userdel 命令

– userdel [-r] 用户名 #-r删除用户信息,并且删除用户的家目录

[root@A ~]# userdel nsd01

[root@A ~]# id nsd01

### 用户查询相关命令：

id //查询用户信息，例如用户归属那个用户组，及uid和gid等。

w //查看谁登录了，并且做了什么，切换的不行，从新登录的才行。

who //查看哪些用户在登录，终端及登录时间，来源主机。

users //仅显示哪些用户在登录。

last //显示的用户曾经加目前登录以及退出登录列表

last log //可以查看是否有用户登录系统了（未启用的账户）

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

## 管理组

前面章节知道用户账户在控制单个用户安全性方面很好，但涉及到共享资源或把用户类型分组时，组概念就出来了。 组权限允许多个用户对系统中的对象(比如文件，目录，设备等)共享一组共用的权限。创建一个用户会为该用户单独创建一个组，这样可以更安全一些。

### groupadd 命令添加组

**组基本信息存放在 /etc/group 文件**

**组密码信息存放在 /etc/gshadow 文件**

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

adminuser:x:1507:

组名:组的密码占位符:组的GID:组的成员列表

– groupadd [选项] 组名

-g：指定新建用户组的gid；

-r：创建系统工作组，系统用户的组ID小于500；

-K：覆盖配置文件“/ect/login.defs”；

-o：允许添加组ID号不唯一的工作组。

groupadd -g 344 bqh

[root@A ~]# groupadd adminuser

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

[root@A ~]# useradd harry

[root@A ~]# useradd Natasha

### gpasswd 命令管理组成员

– gpasswd -a 用户名 组名 #添加用户到组

– gpasswd -d 用户名 组名 #从组内删除用户

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

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

[root@A ~]# id kenji

[root@A ~]# gpasswd -d kenji adminuser

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

### groupmod 命令修改组属性

– groupmod [-g 组ID] [-n 新组名] 组名

groupmod 参数 用户组名

-g： GID 为用户组指定新的组标识号。

-o：与-g选项同时使用，用户组的新GID可以与系统已有用户组的GID相同。

-n：新用户组 将用户组的名字改为新名字

[root@server0 ~]# groupmod -g 1100 market

### groupdel 命令删除组

– groupdel 组名

当Linux创建一个用户,内容改变的文件

[root@A ~]# ls /etc/passwd

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

[root@A ~]# ls /etc/group

[root@A ~]# ls /etc/gshadow

## tar备份与恢复

作用: 1.将零散的文件进行归档到一个文件

2.减小空间的占用

Linux常见的压缩格式及命令工具:

gzip ----> .gz

bzip2 ----> .bz2

xz ----> .xz

### tar 集成备份工具

– -c：创建归档

– -x：释放、解压归档

– -f：指定tar包的名称,必须放在所有选项的最后文件作用取决于前面的参数，若为创建(-c),则-f后为创建的文件的名字(路径)，若为(-x/t),则-f后为待解压/查看的打包压缩文件名

– -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)：指定释放的路径

-P(大写)：保留原文件的绝对路径，即不会拿掉文件路径开始的根目录，则在还原时会覆盖对应路径上的内容

-p(小写)：保留文件的权限和属性，在备份文件时较有用

### 制作压缩包,tar命令格式:

tar 选项 压缩包的名字 被压缩的源文档1 被压缩的源文档2 ......

tar -zcf 压缩包的名字tar.gz 被压缩的源文档1 被压缩的源文档2 ......

tar -jcf 压缩包的名字tar.bz2 被压缩的源文档1 被压缩的源文档2 ......

tar -Jcf 压缩包的名字 tar.xz 被压缩的源文档1 被压缩的源文档2 ......

[root@A ~]# tar -zcf /opt/abc.tar.gz /etc/passwd /etc/shadow /etc/group

[root@A ~]# tar -jcf /opt/test.tar.bz2 /home /etc/passwd

[root@A ~]# tar -Jcf /opt/file.tar.xz /etc/fstab /etc/hosts

[root@A ~]# ls /opt/

**自动识别:**

tar caf 包名 被压缩目录 a 根据报名的后缀自动识别

[root@client50 ~]#tar caf testsdf.tar.gz testsdf

[root@client50 ~]#ls

[root@client50 ~]#file testsdf.tar.gz

testsdf.tar.gz: gzip compressed data, from Unix, last modified: Tue Apr 2 20:49:16 2019

### 释放压缩包,tar命令格式:

tar 选项 压缩包的名字 -C 释放的路径

[root@A ~]# tar -xf /opt/abc.tar.gz -C /mnt/

[root@A ~]# ls /mnt/etc/

[root@A ~]# tar -xf /opt/test.tar.bz2 -C /mnt/

[root@A ~]# ls /mnt/

[root@A ~]# ls /mnt/etc/

[root@A ~]# tar -xf /opt/file.tar.xz -C /mnt/

[root@A ~]# ls /mnt/

### 查看tar包数据内容

[root@A ~]# tar -tf /opt/file.tar.xz

[root@A ~]# tar -tf /opt/abc.tar.gz

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

### 案例2:创建一个备份包

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

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

## NTP时间同步

NTP网络时间协议:Network Time Protocol

– 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

NTP服务器:虚拟机classroom NTP客户机:虚拟机server

软件：chrony 服务名：chronyd

### 1.安装chrony软件与NTP服务器沟通

[root@A ~]# rpm -q chrony

chrony-1.29.1-1.el7.x86\_64

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

### 2.修改主配置文件,指定NTP服务器位置

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

以#开头的位注释行

#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 #指定服务端位置

### 3.重起程序(服务)

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

[root@A ~]# systemctl enable chronyd #设置服务开机自起

daemon:守护神;（希腊神话中）半人半神的精灵;[计]守护进程

### 4.验证时间同步

[root@A ~]# date

2018年 12月 05日 星期三 17:12:42 CST

[root@A ~]# date -s '2000-10-1 10:11'

2000年 10月 01日 星期日 10:11:00 CST

[root@A ~]# date

# 系统管理员ADMIN06

## 文档基本权限--r、w、x

### 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权限常见操作: cat less tail head 查看内容

w权限常见操作: vim > >>

x权限创建操作: 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

### ls -l 命令查看权限

– ls -ld 文件或目录...

文档开头字目意思：以 - 开头:文本文件

以 d 开头:目录

以 l 开头:快捷方式

以c开头:代表字符型设备

以b开头:代表块设备

以n开头:代表网络设备

[root@server0 ~]# ls -l /etc/passwd

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

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

drwxr-xr-x.:2:root:root

1以—开头：文本文件 以d开头：目录 以l开头：快捷方式

2所有者权限：rwx 所属组权限：r-w 其他人权限：r-w

3硬连接：2

4属主（所有者）

5属组（所属组）

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

### chmod 命令设置基本权限

– chmod [-R] 归属关系+/-/=权限类别 文档...

[-R] :递归修改权限

[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 o=--- /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

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

### Linux判断用户具备的权限:

匹配及停止

1.判断用户对于文档,属于什么身份 所有者>所属组>其他人

2.查看文档,相应角色的权限

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

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

### chmod练习：

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文件内容

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/

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

## 文档归属--属主、属组

### chown 命令设置文档归属

– chown [-R] 属主 文档...

– chown [-R] :属组 文档...

– chown [-R] 属主:属组 文档...

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

[root@server0 /]# groupadd tarena

[root@server0 /]# chown lisi:tarena /nsd03 #修改nsd03目录的所有者为lisi、所属组为tarena

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

[root@server0 /]# useradd zhangsan

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

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

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

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

用户加入组不会立即生效，需要重新登入，判断等级：属主>属组>其他人，匹配即停止

### chown练习

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 /]# useradd gelin02

[root@server0 /]# groupadd tarena

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

2）使用户gelin01对此目录具有rwx权限,除属主与属组之外的人，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 /]# gpasswd -a gelin02 tarena

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

## Set GID附加权限(g目录的特殊权限)

* 附加在属组的 x 位上
* 属组的x权限标识位会变s,如原权限有x权限s为小写，如原权限无x权限s为大写
* 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
* 传递所属组身份(让子文档自动继承父目录的所属组身份)
* 命令格式：chmode g+s 目录

### 命令格式：chmod g+s 目录

[root@server0 /]# mkdir /nsd07

[root@server0 /]# chown :tarena /nsd07 #修改所属组身份

[root@server0 /]# ls -ld /nsd07

[root@server0 /]# mkdir /nsd07/abc01 #在nsd07目录里面新建abc01目录

[root@server0 /]# ls -ld /nsd07/abc01 #查看abc01目录权限

[root@server0 /]# chmod g+s /nsd07 #设置Set GID权限

[root@server0 /]# ls -ld /nsd07 #查看Set GID 权限设置成功

[root@server0 /]# mkdir /nsd07/abc02 #在nsd07目录里面新建abc02目录

[root@server0 /]# ls -ld /nsd07/abc02 #查看abc02目录权限，自动继承nsd07目录权限

[root@server0 /]# mkdir /nsd07/abc02/test #深层次的文档

[root@server0 /]# ls -ld /nsd07/abc02/test #同样继承nsd07的权限

### 练习:配置附加权限

创建一个共用目录 /home/admins,要求如下:

– 此目录的组所有权是 adminuser

– adminuser 组的成员对此目录有读写和执行的权限,

-- 除此以外的其他所有用户没有任何权限(root用户能够访问系统中的所有文件和目录)

– 在此目录中创建的文件,其组的所有权会自动设置为属于 adminuser 组

[root@server0 /]# mkdir /home/admins

[root@server0 /]# ls -ld /home/admins

[root@server0 /]# groupadd adminuser

[root@server0 /]# chown :adminuser /home/admins

[root@server0 /]# ls -ld /home/admins

[root@server0 /]# chmod g+w /home/admins

[root@server0 /]# ls -ld /home/admins

[root@server0 /]# chmod o=--- /home/admins

[root@server0 /]# ls -ld /home/admins

[root@server0 /]# chmod g+s /home/admins

[root@server0 /]# ls -ld /home/admins

[root@server0 /]# touch /home/admins/1.txt

[root@server0 /]# ls -l /home/admins/1.txt

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

## Set UID附加权限（u适用于可执行文件）

* 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

### 命令格式：chmod u+s 文件名

[root@server0 ~]# ls /usr/bin/mkdir

/usr/bin/mkdir

[root@server0 ~]# /usr/bin/mkdir /opt/nsd02

[root@server0 ~]# ls /opt/

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

[root@server0 ~]# /usr/bin/hahadir /opt/nsd03

[root@server0 ~]# ls /opt/

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

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

[root@server0 ~]# su - student

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

[student@server0 ~]$ ls -l

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

[student@server0 ~]$ ls -l

[student@server0 ~]$ exit

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

## Sticky Bit附加权限（t适用开放w的目录）

* 附加在其他人权限的x位上，变t
* 适用于开放w权限的目录，可以组织用户滥用w写入权限，
* 禁止操作别人的文档
* 防止误覆盖，误删除操作

### 命令格式：Chmod o+t 目录

[root@server0 ~]# mkdir /home/public

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

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

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

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

drwxrwxrwt. 2 root root 6 12月 7 10:16 /home/public

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

## acl访问控制列表

acl策略的作用

• 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

– 大多数挂载的EXT3/4、XFS文件系统默认已支持

[root@server0 /]# mkdir /nsd10

[root@server0 /]# chmod o=--- /nsd10

[root@server0 /]# ls -ld /nsd10

[root@server0 /]# su - lisi

[lisi@server0 ~]$ cd /nsd10 #切换失败

-bash: cd: /nsd10: Permission denied

[lisi@server0 ~]$ exit

[root@server0 /]# setfacl -m u:lisi:rx /nsd10 #设置ACL策略

[root@server0 /]# getfacl /nsd10 #查看ACL策略

[root@server0 /]# su - lisi

[lisi@server0 ~]$ cd /nsd10 #切换成功

[lisi@server0 nsd10]$ pwd

[lisi@server0 nsd10]$ exit

### getfacl查看、setfacl设置命令

– getfacl 文档...

– 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] –x u:用户名 文档... #删除指定的ACL策略

– setfacl [-R] -b 文档... #清空所有ACL策略

[root@server0 /]# mkdir /nsd11

[root@server0 /]# setfacl -m u:lisi:rwx /nsd11

[root@server0 /]# setfacl -m u:zhangsan:rwx /nsd11

[root@server0 /]# setfacl -m u:dc:rx /nsd11

[root@server0 /]# useradd harry

[root@server0 /]# setfacl -m u:harry:rwx /nsd11

[root@server0 /]# getfacl /nsd11

[root@server0 /]# setfacl -x u:dc /nsd11 #删除指定的ACL

[root@server0 /]# getfacl /nsd11

[root@server0 /]# setfacl -x u:lisi /nsd11

[root@server0 /]# getfacl /nsd11

[root@server0 /]# setfacl -b /nsd11 #清除所有的ACL

[root@server0 /]# getfacl /nsd11

### ACL设置黑名单

[root@server0 /]# mkdir /public

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

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

[root@server0 /]# setfacl -m u:lisi:--- /public #lisi对public无任何权限

[root@server0 /]# getfacl /public

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

请实现lisi用户,可以查看/etc/shadow文件内容,您有几种方法?

1.利用其他人 chmod o+r /etc/shadow

2.利用所属组 将lisi加入文件所属组中

chmod g+r /etc/shadow

3.利用所有者 将lisi设置为文件的所有者

chown lisi /etc/shadow

chmod u+r /etc/shadow

4.利用ACL setfacl -m u:lisi:r /etc/shadow

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

## LDAP认证

### 什么是LDAP?

• 轻量级目录访问协议

– Lightweight Directory Access Protocol

只适用于纯redhat系统

– 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、......用户的集中管理

LDAP用户:网络用户,由网络中LDAP服务器统一提供用户信息

LDAP服务器:虚拟机classroom.example.com

LDAP客户端:虚拟机server

### 1.客户端安装软件sssd,

sssd用于客户端与LDAP服务器沟通

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

[root@server0 /]# rpm -q sssd

### 2.图形工具authconfig-gtk进行配置

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

[root@server0 /]# authconfig-gtk #启动图形工具

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

### 3.重起sssd服务

[root@server0 ~]# systemctl restart sssd #重起服务

[root@server0 ~]# systemctl enable sssd #设置开机自起服务

### 4.验证:

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

[root@server0 ~]# id ldapuser0

[root@server0 ~]# id ldapuser1

## 家目录漫游

将虚拟机classroom上用户的家目录,出现在虚拟机server

此处是利用NFS共享,classroom已经部署好NFS,只要使用即可,如何部署在下面有

### 什么是NFS共享

• NFS：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服务器:虚拟机classroom

客户端虚拟机server:访问共享

### 查看NFS资源

– showmount -e [服务器地址]

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

Export list for classroom.example.com:

/home/guests 172.25.0.0/255.255.0.0

• 挂载NFS共享目录

– mount 服务器地址:目录路径 本地挂载点

]# mkdir /home/guests

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

]# ls /home/guests

]# su - ldapuser0

# 系统管理员ADMIN07

## 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

* 创建任务计划：crontab –e –u 用户名 #也可修改
* 查看任务计划：crontab –l – u 用户名
* 清除任务计划：crontab –r – u 用户名

任务计划存放目录：/var/spool/cron #不同用户有不同文件可用vim直接修改，cat查看

如何编写crontab任务记录

– 分 时 日 月 周 任务命令行(绝对路径)

30 8 \* \* \* #每天8:30执行命令

0 23 \* \* 5 #每个星期五23:00执行命令

0 23 \* \* 7 #每个星期天23:00执行命令

0 8 1 \* \* #每个月的1日8:00执行命令

0 8 \* \* 1-5 #每个星期1—星期5的8:00执行命令

0 8 \* \* 1,6,7 #每个星期1、6、7的8:00执行命令

0 8 1 \* 1 #每月1号和每个周一8:00也会执行命令

0 \*/2 \* \* \* #每2个小时执行一次

\*/4 \* \* \* \* #每4分钟执行一次

星号\*：匹配范围内任意时间

逗号,：分隔多个不连续的时间点

横杠-：指定连续时间范围

/n：指定时间频率,每n ..

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

每分钟记录当前的系统时间,写入到/opt/time.txt

[root@server0 ~]# date >> /opt/time.txt #原理就是date的结果追加重定向到文本里面

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

[root@server0 ~]# crontab -e -u root #新建任务计划

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

[root@server0 ~]# ls /var/spool/cron/

[root@server0 ~]# cat /var/spool/cron/root #查看生成的任务文件

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

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

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

## grep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' 被查找文本文件...

– 命令 | grep [选项] '匹配模式' #结合管道方式

### • 常用命令选项

– -v：取反匹配

– -i：忽略大小写

[root@server0 ~]# grep ROOT /etc/passwd #在passwd文件中查找含有大写ROOT行

[root@server0 ~]# grep -i ROOT /etc/passwd #在passwd中查找含有root行，不限大小写

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

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

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

[root@server0 ~]# grep -v root /etc/passwd #在passwd中查找除了含有root其他所有

### • 常用的匹配模式

– word 包含字符串word

– ^word 以字符串word开头

– word$ 以字符串word结尾

– ^$ 表示空行

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

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

[root@server0 ~]# cat /etc/default/useradd

[root@server0 ~]# grep ^$ /etc/default/useradd

[root@server0 ~]# grep -v ^$ /etc/default/useradd

**例：显示文件有效信息(去除以#号开头的注释,去除空行)**

]# grep -v ^# /etc/login.defs

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

**例：将/etc/login.defs文件有效信息,写入到/opt/log.txt**

]# grep -v ^# /etc/login.defs

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

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

]# cat /opt/log.txt

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

## find查找文件

命令格式：find [目录] [条件1] [-a|-o] [条件2] ...

常用条件表示:

-type 类型(f 文本文件、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

### 案例

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

/boot/grub/menu.lst

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

/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 #查找是文本文件

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

### -name "文档名称"

[root@server0 ~]# find /boot/ -name "vm\*"

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

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

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

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

[root@server0 ~]# touch /root/nsd1811.txt

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

[root@server0 ~]# find /root/ -name "nsd\*" -type f #查找nsd开头的，且类型是文本文件

[root@server0 ~]# find /root/ -name "nsd\*" -type d #查找nsd开头的，且类型目录的文档

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

### -size +|- 文件大小(k、M、G)

]# find /boot/ -size +300k #查找 大于300k

]# find /boot/ -size +10M #查找 大于10M

]# ls -lh /boot/initramfs-3.10.0-123.el7.x86\_64.img

]# find /boot/ -size -10M #查找 小于10M

### -user 根据用户名

]# find / -user student #查找 文档的所有者为student用户

### -i name根据名称查找,忽略大小写

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

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

### -group根据所属组

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

### -maxdepth限制目录查找的深度(最大层数)

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

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

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

### -mtime根据文件修改时间

-mtime 所有的时间都是过去时间

+10 :查找10天之前创建或修改的文档

-10 :查找最近10天之内创建或修改的文档

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

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

### 使用find命令的 -exec 操作

命令格式： find .. .. -exec 处理命令 {} \;

– 优势:以 {} 代替每一个结果,逐个处理,遇 \; 结束

案例:查找并处理文件

• 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

]# mkdir /root/findfiles

]# find / -user student -type f -exec cp {} /root/findfiles/ \;

]# ls -A /root/findfiles/

案例：查找boot目录里面大于10M的文件，并且复制到opt目录里面

]# find /boot/ -size +10M

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

]# ls /opt

# 应用管理ENGINEER01

## 分区规划及使用

扇区默认的大小为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

## lsblk查看硬盘

[root@server0 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 10G 0 disk

└─vda1 253:1 0 10G 0 part /

vdb 253:16 0 10G 0 disk

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

名字开头为vd：虚拟化硬盘

名字开头为hd：IDE硬盘设备

名字开头为sd：SCSI硬盘设备

## 交互式fdisk分区工具

### MBR模式（老模式）：主引导记录

* 最大支持的空间2.2Tb
* 分区类型: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)
* 扩展分区不能格式化,不能存储数据

fdisk分区工具用来建立msdos分区方案，其交互模式中的主要指令如下：

m：列出指令帮助

p：查看当前的分区表信息

n：新建分区

d：删除分区

t：更改分区标识

q：放弃分区更改并退出

w：保存对分区表所做的更改

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

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

p 查看分区表

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

d 删除分区

w 保存并退出

[root@server0 ~]# lsblk

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

### mkfs工具格式化文件系统

格式化文件系统：存储的数据的规则

– mkfs.ext3 分区设备路径 适合小文件,rhel5

– mkfs.ext4 分区设备路径 适合小文件,rhel6

– mkfs.xfs 分区设备路径 适合大文件,rhel7

– mkfs.vfat -F 32 分区设备路径

--- mkfs.xfs - f /dev/vdb2 #选项-f强制格式化

[root@server0 ~]# mkfs.ext4 /dev/vdb1 #格式化文件系统为ext4

### blkid查看文件系统类型

[root@server0 ~]# blkid /dev/vdb1

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

[root@server0 ~]# blkid /dev/vdb2

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

mkfs.xfs: /dev/vdb2 appears to contain an existing filesystem (ext4).

mkfs.xfs: Use the -f option to force overwrite.

[root@server0 ~]# mkfs.xfs -f /dev/vdb2 #强制格式化

### 分区挂载使用

[root@server0 ~]# mkdir /mypart1

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

### df 查看已挂载的设备

[root@server0 ~]# df -h #显示正在挂载设备的使用情况，以易读方式显示

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

mount: 挂载点 /mypart2 不存在

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

[root@server0 ~]# df -h | tail -2

### 开机自动挂载设备

• 配置文件 /etc/fstab 的记录格式

– 设备路径 挂载点 类型 参数 备份标记 检测顺序

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

......

设备路径 挂载点 类型 参数 备份标记 检测顺序

/dev/vdb1 /mypart1 ext4 defaults 0 0

/dev/vdb2 /mypart2 xfs defaults 0 0

[root@server0 ~]# umount /mypart1

[root@server0 ~]# umount /mypart2

[root@server0 ~]# df -h

[root@server0 ~]# mount -a

检测/etc/fstab开机自动挂载配置文件,格式是否正确

检测/etc/fstab中,书写完成,但当前没有挂载的设备,进行挂载

[root@server0 ~]# df -h

### 综合分区

最终有3个主分区，分别为2G、1G、2G

创建扩展分区 ---> 两个逻辑分区，分别为1G、1G

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

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+1G

n 创建逻辑分区----->起始回车------>结束+1G

p 查看分区表

w 保存并退出

[root@server0 ~]# lsblk

[root@server0 ~]# partprobe #刷新 分区表信息 识别新的分区

[root@server0 ~]# lsblk

### 新装系统分区

linux至少只要一个根分区就行,

物理服务器:

/boot分区:500M

/home:看情况是否需要单独分区,,如果各个用户需要用来存储数据,可分

/var:一般单独分成分区,日志等可变文件,可能会自增长,

/:根分区,

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

### 命令总结

lsblk 1.查看本机硬盘设备

fdisk /dev/vda 2.划分新的分区

partprobe 3.刷新 新的分区表

mkfs.ext4 /dev/vdb1 4.格式化文件系统

mkfs.xfs /dev/vdb2 4.格式化文件系统

blkid /dev/vdb1 5.查看文件系统

mount 6.手工挂载使用

/etc/fstab 7.配置开机自动挂载

mount -a 8.检测fstab文件是否配置正确，将配置了的未挂载的挂载使用

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

### 案例：新添加一块80G分区:

1.关闭虚拟机server

[root@server0 ~]# poweroff

Connection to 172.25.0.11 closed by remote host.

Connection to 172.25.0.11 closed.

[root@room9pc01 ~]#

2.利用图形添加80G硬盘

[root@room9pc01 ~]# gos

[root@server0 ~]# lsblk

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

综合分区:

划分3个主分区,分别为10G

划分扩展分区,在扩展分区之上,划分2个10G的逻辑分区

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

p 查看分区表

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

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

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

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

p 查看分区表

w 保存并退出

[root@server0 ~]# lsblk

/dev/sda5表示的含义?

第一块SCSI设备,第五个分区

第一块SCSI设备,第一个逻辑分区

## LVM逻辑卷

作用: 1.整合分散的空间 2.空间可以按照需求扩大

零散空闲存储 ---- 整合的虚拟磁盘 ---- 虚拟的分区

物理卷 (PV)Physical Volume

卷组 (VG)Volume Group

逻辑卷 (LV)Logical Volume

将众多的物理卷(PV),组建成卷组(VG),从卷组中划分出逻辑卷(LV)

首先将磁盘或分区转换成物理卷PV,然后,将1到多个PV组合成卷组VG,最后,从VG上划分逻辑卷LV,LV就像是普通分区一样,格式化,挂在,LV不够,可以实现在线扩容.

### 一：创建逻辑卷

#### 创建物理卷（pv）（可省略）

命令格式：]#pvcreate 分区目录1 分区目录2

[root@server0 ~]# pvcreate /dev/vdc1 /dev/vdc2 #将分区转换成PVN

[root@server0 ~]#pvs #查看物理卷(pv)

#### 创建卷组（vg）

命令格式: vgcreate [选项] 卷组名 设备路径......

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

[root@server0 ~]# vgs #查看卷组信息

[root@server0 ~]# pvs #查看物理卷信息

#### 创建逻辑卷（lv）

命令格式:lvcreate -L 大小 -n 逻辑卷名称 基于的卷组

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

Logical volume "vo" created

[root@server0 ~]# lvs #查看逻辑卷信息

[root@server0 ~]# vgs #查看卷组信息

#### 使用逻辑卷

[root@server0 ~]# ls /dev/systemvg/vo

[root@server0 ~]# ls -l /dev/systemvg/vo #查看逻辑卷

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

[root@server0 ~]# blkid /dev/systemvg/vo #查看文件格式

[root@server0 ~]# vim /etc/fstab #设置开机自挂载

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

[root@server0 ~]# mount –a #挂载

mount: 挂载点 /mylv 不存在

[root@server0 ~]# mkdir /mylv

[root@server0 ~]# mount -a

[root@server0 ~]# df –h #查看挂载后的情况

### 二 逻辑卷的扩展

#### 1.卷组有足够的剩余空间

A:扩展逻辑卷的空间

[root@server0 ~]# lvs

[root@server0 ~]# vgs

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

[root@server0 ~]# lvs

[root@server0 ~]# df -h

B:扩展文件系统（格式化）

resize2fs:扩展ext4文件系统

xfs\_growfs:扩展xfs文件系统

[root@server0 ~]# df -h

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

[root@server0 ~]# df -h

#### 2.卷组没有足够的剩余空间

A:扩展卷组（加分区）

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

Physical volume "/dev/vdc3" successfully created

Volume group "systemvg" successfully extended

[root@server0 ~]# vgs

B:扩展逻辑卷的空间

[root@server0 ~]# lvs

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

[root@server0 ~]# lvs

[root@server0 ~]# df -h

C:扩展文件系统

[root@server0 ~]# df -h

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

[root@server0 ~]# df -h

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

了解内容:

逻辑卷空间可以进行缩小

ext4文件系统支持缩小

xfs文件系统不支持缩小

### PE卷组划分空间的单位

PE：默认4M，逻辑卷增加划分空间是按照PE单元个数来增减，例如：按照默认的4M大小，以4M为一个单位增减，增减都为4的倍数。

#### 显示卷组详细信息

[root@server0 ~]# vgdisplay systemvg

PE Size 4.00 MiB

#### 创建卷组的时候设置PE大小

– vgcreate -s PE大小 卷组名 空闲分区...

#### 修改卷组的时候设置PE大小

– vgchange -s PE大小 卷组名

]#vgchange -s 1M systemvg

]#lvcreate -L 250M –n lvtest02 systemvg #直接以大小创建逻辑卷

]#lvcreate -l 50 -n lvtest02 systemvg #以50个PE单元创建逻辑卷

#### 例题：

请创建一个逻辑卷为lvtest01 ,大小为250M

[root@server0 ~]# vgchange -s 1M systemvg #修改PE值为1M，250不是4的倍数

Volume group "systemvg" successfully changed

[root@server0 ~]# lvcreate -L 250M -n lvtest02 systemvg

Logical volume "lvtest02" created

[root@server0 ~]# lvs

基于刚建立的 2000MiB 分区构建新的 LVM 存储，新的逻辑卷命名为 database,大小为50个物理扩展单元(Physical Extent)，属于 datastore 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16MiB，

[root@server0 ~]# vgcreate -s 16M datastore /dev/sdd1

[root@server0 ~]# vgdisplay datastore #显示卷组详细信息

PE Size 16.00 MiB

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

[root@server0 ~]#

### 逻辑卷删除

删除的顺序:

首先删除逻辑卷,再删除卷组,最后删除物理卷(可以不删除物理卷)

#### 1.先卸载逻辑卷

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

Logical volume systemvg/vo contains a filesystem in use.

[root@server0 ~]# umount /mylv/

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

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

Logical volume "vo" successfully removed

[root@server0 ~]# lvs

[root@server0 ~]# lvremove /dev/systemvg/lvtest01

Do you really want to remove active logical volume lvtest01? [y/n]: y

Logical volume "lvtest01" successfully removed

[root@server0 ~]# lvremove /dev/systemvg/lvtest02

Do you really want to remove active logical volume lvtest02? [y/n]: y

Logical volume "lvtest02" successfully removed

[root@server0 ~]# lvs

#### 2.再删除卷组

(没有任何一个逻辑卷,基于此卷组创建)

[root@server0 ~]# vgs

[root@server0 ~]# vgremove systemvg

Volume group "systemvg" successfully removed

[root@server0 ~]# vgs

No volume groups found

[root@server0 ~]#

# 应用管理ENGINEER02

## Shell脚本基础

脚本:可以执行的一个文件,实现某种功能

规范Shell脚本的一般组成

• #! 环境声明

• # 注释文本

• 可执行代码

案例 编写一个面世问候 /root/hello.sh 脚本显示出一段话 Hello World

1)输出当前红帽系统的版本信息

2)输出当前使用的内核版本

3)输出当前系统的主机名

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

#! /bin/bash

echo hello World

cat /etc/redhat-release

uname -r

hostname

ifconfig | head -2

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

[root@server0 ~]# ls -l /root/hello.sh

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

## 重定向输出

> :只收集前面命令的正确输出

2>:只收集前面命令的错误输出

&>:收集前面命令的正确输出与错误输出

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

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

[root@server0 ~]# cat /a

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

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

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

[root@server0 ~]# cat /opt/1.txt /a 2> /opt/b.txt

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

[root@server0 ~]# cat /opt/1.txt /a &> /opt/b.txt

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

### 案例:书写一个创建用户的脚本,并且设置密码

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

#!/bin/bash

a=nsd04

useradd $a &> /dev/null

echo 用户$a创建成功

echo 123 | passwd --stdin $a &> /dev/null

echo 用户$a密码设置成功

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

[root@server0 ~]# /root/user.sh

### 黑洞设备:/dev/null

## 变量

为了让脚本的增加灵活度,适应多变的环境,使用变量

变量=容器 以不变的名称,存放可以变化的值

定义变量:变量名=存放的值

定义/赋值变量

### 设置变量时的注意事项

– 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

– 等号两边不要有空格

### 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=1000

[root@server0 ~]# echo $a

1000

[root@server0 ~]# echo ${a}

1000

[root@server0 ~]# a=rhel

[root@server0 ~]# echo $a7

#输出为空

[root@server0 ~]# echo ${a}7

rhel7

### 根据变量的用途不同区分

#### 环境变量

环境变量: 变量名一般都大写,用来设置用户/系统环境，由系统定义完成,由系统赋值完成

例如：USER=当前系统登陆的用户名

[root@server0 ~]# echo $USER

root

[root@server0 ~]# su - student

[student@server0 ~]$ echo $USER

student

[student@server0 ~]$ exit

logout

[root@server0 ~]# echo $USER

root

[root@server0 ~]#

#### 位置变量

位置变量: bash内置,存储执行脚本时提供的命令行参数，由系统定义完成,由系统赋值完成表示为 $n,n为序号 $1、$2、.. .. ${10}、${11}、.. ..

[root@server0 ~]# cat /root/user.sh

#!/bin/bash

useradd $1 &> /dev/null

echo 用户 $1 创建成功

echo 123 | passwd --stdin $1 &> /dev/null

echo 用户 $1 密码设置成功

[root@server0 ~]# /root/user.sh natasha

[root@server0 ~]# vim /root/1.sh

#!/bin/bash

cat -n $1 | head -$2

[root@server0 ~]# /root/1.sh /etc/passwd 3

#### read作用:

为了降低脚本的使用难度,提高用户的体验度,采用交互式传递参数方法

1.产生交互

2.自动记录用户在键盘上的输入

3.自动将用户在键盘上的输入,赋值给一个变量储存

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

#!/bin/bash

read -p '请输入您要创建的用户名:' a

useradd $a &> /dev/null

echo 用户 $a 创建成功

echo 123 | passwd --stdin $a &> /dev/null

echo 用户 $a 密码设置成功

[root@server0 ~]# /root/user.sh

#### 预定义变量

预定义变量: bash内置,可直接调用的特殊值,不能直接修改由系统定义完成,由系统赋值完成

$# 已加载的位置变量的个数,是传给脚本的参数个数

$\* 所有位置变量的值

$@ 跟$\*类似，是传给脚本的所有参数的列表,但是可以当作数组用

$? 程序退出后的状态值,0表示正常,其他值异常

$$ Shell本身的PID（ProcessID，即脚本运行的当前进程ID号）

$! Shell最后运行的后台Process的PID(后台运行的最后一个进程的进程ID号)

$0 Shell本身的文件名

[root@server0 ~]# cat /root/2.sh

#!/bin/bash

echo $1

echo $2

echo $3

echo 您一共输入了 $# 个位置变量

echo 它们的值分别为 $\*

[root@server0 ~]# /root/2.sh dc tc dz tz

#### 自定义变量:

自定义变量: 用户自主设置、修改及使用

## 单引号 ' '

取消特殊字符的意义,原样输出

[root@server0 ~]# a=100

[root@server0 ~]# echo $a

100

[root@server0 ~]# echo '$a'

$a

$加变量名可以取变量的值 ，比如：

　[root@localhost ~]# echo '$PWD'

　$PWD

　[root@localhost ~]# echo "$PWD"

　/root

## 反撇号 ` `与$( )

将命令的输出,直接参数其他命令运行

[root@server0 ~]# rm -rf /opt/\*

[root@server0 ~]# mkdir /opt/nsd-`date +%F`

[root@server0 ~]# ls /opt/

nsd-2019-02-24

[root@server0 ~]# mkdir /opt/abc-$(date +%F)

[root@server0 ~]# ls /opt/

abc-2019-02-24 nsd-2019-02-24

[root@server0 ~]# mkdir /opt/`hostname`-$(date +%F)

[root@server0 ~]# ls /opt/

abc-2019-02-24 nsd-2019-02-24 server0-2019-02-24

## 条件测试及选择

条件测试

[ 测试表达式 ] #每一部分之间都要有空格

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,并且为目录才为真

-f:文档存在,并且为文件才为真

-r:文档存在,并且对其有读取权限为真

-w:文档存在,并且对其有写入权限为真

-x:文档存在,并且对其有执行权限为真

[root@server0 ~]# [ -e /etc ]

[root@server0 ~]# echo $?

0

[root@server0 ~]# [ -e /abc ]

[root@server0 ~]# echo $?

1

[root@server0 ~]# [ -f /etc/passwd ]

[root@server0 ~]# echo $?

0

[root@server0 ~]# [ -f /etc ]

[root@server0 ~]# echo $?

1

[root@server0 ~]# [ -d /etc ]

[root@server0 ~]# echo $?

0

### 比较整数大小

(带e的都有等于二字,g是大于,l是小于)

-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 $?

0

[root@server0 ~]# [ 1 -gt 2 ]

[root@server0 ~]# echo $?

1

[root@server0 ~]# [ 1 -ge 1 ]

[root@server0 ~]# echo $?

0

[root@server0 ~]# [ 10 -lt 11 ]

[root@server0 ~]# echo $?

0

### 字符串比对

==:相等一致为真 !=:不相等为真

[root@server0 ~]# [ student == root ]

[root@server0 ~]# echo $?

1

[root@server0 ~]# [ root == root ]

[root@server0 ~]# echo $?

0

[root@server0 ~]# [ $USER == student ]

[root@server0 ~]# echo $?

1

[root@server0 ~]# [ $USER != student ]

[root@server0 ~]# echo $?

## if选择结构

### if单分支处理

if 条件测试

then

命令序列

fi

### if双分支处理

if [条件测试];then

命令序列xx

else

命令序列yy

fi

[root@server0 ~]# vim /root/if01.sh

#!/bin/bash

if [ $1 -eq 1 ];then

echo hello

else

echo hi

fi

[root@server0 ~]# /root/if01.sh 1

[root@server0 ~]# /root/if01.sh 2

[root@server0 ~]# /root/if01.sh 3

#### if单分支案例: 判断IP地址的脚本

利用read 获取用户的输入IP地址 进行判断:

如果本机与用户的输入IP地址可以通信,则输出 可以通信

如果本机与用户的输入IP地址不可以通信,则输出 无法通信

[root@server0 ~]# vim /root/if02.sh

#!/bin/bash

read -p '请输入您要测试的IP地址:' ip

ping -c 2 $ip &> /dev/null

if [ $? -eq 0 ];then

echo $ip 可以通信

else

echo $ip 无法通信

fi

[root@server0 ~]#

### 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cc

.......

else

命令序列zz

fi

#### if多分支案例:书写考试成绩脚本

利用read 获取用户的输入成绩：如果成绩大于等于90,则输出 优秀，如果成绩大于等于80,则输出：良好，如果成绩大于等于70,则输出：一般，如果成绩大于等于60,则输出：合格

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/num.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

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

## for循环结构

for循环结构:可以重复执行一个操作，根据变量的不同取值,重复执行xx处理

### 遍历/列表式循环

for 变量名 in 值列表

do

命令序列

done

for a in zhangsan lisi wangwu

do

useradd $a

done

[root@server0 ~]# cat /root/for01.sh

#! /bin/bash

for a in zhangsan lisi wangwu dc tc dz tz

do

useradd $a &> /dev/null

echo $a 创建成功

done

[root@server0 ~]# /root/for01.sh

### for循环中,变量的值可以不参与循环执行的动作

[root@server0 ~]# cat /root/for01.sh

#!/bin/bash

for a in zhangsan lisi wangwu dc tc dz tz

do

echo hello world

done

[root@server0 ~]#

### 造数工具:{起始点..结束点}

例如：{1..50}生成1到50之间所有的数字

[root@server0 ~]# cat /root/for01.sh

#!/bin/bash

for a in {1..15}

do

echo hello world

done

[root@server0 ~]#

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

## 案例1：Shell脚本判断用户是否存在

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 '请输入您测试是的用户名:' a

id $a &> /dev/null

if [ $? -eq 0 ];then

echo 用户存在

id $a

else

useradd $a

echo 用户创建成功

fi

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

#!/bin/bash

read -p '请输入您测试是的用户名:' a

useradd $a &> /dev/null

if [ $? -eq 0 ];then

echo 用户创建成功

else

echo 用户存在

id $a

fi

## 案例2:编写一个判断输入的内容脚本

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

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

#!/bin/bash

if [ $# -eq 0 ];then #匹配用户没有输入参数

echo '/root/foo.sh redhat|fedora' >&2 #转变为错误输出

exit 23 #程序退出返回值

elif [ $1 == redhat ];then

echo fedora

elif [ $1 == fedora ];then

echo redhat

else

echo '/root/foo.sh redhat|fedora' >&2 #转变为错误输出

exit 24 #程序退出返回值

fi

[root@server0 ~]#

## 案例3: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:

http://classroom.example.com/pub/materials/userlist

[root@server0 ~]# vim /root/bachusers

#!/bin/bash

if [ $# -eq 0 ];then

echo 'Usage: /root/batchusers' >&2 #将输出变为错误输出

exit 2

elif [ -f $1 ];then #判断命令行参数是否为文件

for a in $(cat $1) #将命令的输出作为列表

do

useradd -s /bin/false $a &> /dev/null

echo $a创建成功

done

else

echo 'Input file not found' >&2

exit 3

fi

[root@server0 ~]#

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

# 应用管理ENGINEER03

## 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

– 集成到Linux内核(2.6及以上)中运行

– RHEL7基于SELinux体系针对用户、进程、目录和文件

提供了预设的保护策略,以及管理工具

### SELinux运行模式的切换

• SELinux的运行模式

enforcing(强制)、

permissive(宽松)

disabled(彻底禁用) 变成disabled状态,必须配置/etc/selinux/config文件并重起系统

• 切换运行模式

– 临时切换:setenforce 1|0 1为enforcing模式 0为permissive模式

– 查看当前模式:getenforce

– 固定配置:/etc/selinux/config 文件

虚拟机server

[root@server0 ~]# vim /etc/selinux/config #修改固定配置文件

SELINUX=permissive

[root@server0 ~]# getenforce #查看SELinux当前模式

[root@server0 ~]# setenforce 0 #修改当前模式为permissive

[root@server0 ~]# getenforce

虚拟机desktop

[root@desktop0 ~]# vim /etc/selinux/config

SELINUX=permissive

[root@desktop0 ~]# getenforce

[root@desktop0 ~]# setenforce 0

[root@desktop0 ~]# getenforce

## 防火墙策略基本使用

虚拟机server

一 搭建基本Web服务

1.安装httpd软件

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

2.重起httpd服务

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# systemctl enable httpd

3.本机访问测试

[root@server0 ~]# firefox 172.25.0.11

4.书写一个页面文件

默认页面文件存放的路径:/var/www/html

默认网页文件名称: index.html

[root@server0 ~]# cat /var/www/html/index.html

<marquee><font color=red><h1>NSD1811 阳光明媚!!

滚动 字体颜色为红色 最大标题字体

[root@server0 ~]# firefox 172.25.0.11

二 搭建基本FTP服务(文件传输协议)

1.安装vsftpd软件

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

2.重起vsftpd服务

[root@server0 ~]# systemctl restart vsftpd

[root@server0 ~]# systemctl enable vsftpd

3.本机访问测试

默认共享数据的路径:/var/ftp

[root@server0 ~]# firefox ftp://172.25.0.11

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

### 防火墙策略管理

作用:隔离过滤，服务名：firewalld

分为：硬件防火墙和软件防火墙

### RHEL7的防火墙体系

• 系统服务:firewalld

• 管理工具:firewall-cmd命令、firewall-config图形

[root@server0 ~]# systemctl restart firewalld #重启防火墙

[root@server0 ~]# firewall-config &

### 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 dhcp ping等少数几个服务

– trusted:允许任何访问

– block:阻塞任何来访请求(明确拒绝,会给回应)

– drop:丢弃任何来访的数据包(直接丢弃,不给回应,节省资源)

### 防火墙判定的规则

1.查看客户端请求中,源IP地址然后查阅自己那个区域,有该IP地址的规则,则进入该区域

2.进入默认区域：root可指定哪个区域作为默认区域，如没修改默认是public区域

数据包: 源IP地址 目标IP地址 数据

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

### 防火墙查看、修改、配置实例

虚拟机server

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

虚拟机desktop

]# ping -c 2 172.25.0.11 #可以通信

虚拟机server

]# firewall-cmd --set-default-zone=block #修改默认防火墙区域

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

虚拟机desktop

]# ping -c 2 172.25.0.11 #不可以通信,有回应

虚拟机server

]# firewall-cmd --set-default-zone=drop #修改默认防火墙区域

]# firewall-cmd --get-default-zone

虚拟机desktop

]# ping -c 2 172.25.0.11 #不可以通信,没有回应

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

#### 临时在public区域添加服务或协议

以public区域为例，其他区域也可添加或删除协议，永久添加（--permanent）

虚拟机server

]# firewall-cmd --set-default-zone=public #修改默认区域

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

]# firewall-cmd --zone=public --list-all #查看public区域规则

虚拟机desktop

]# firefox 172.25.0.11 #无法访问

]# firefox ftp://172.25.0.11 #无法访问

虚拟机server

]# firewall-cmd --zone=public --add-service=ftp #添加协议

]# firewall-cmd --zone=public --add-service=http #添加协议

]# firewall-cmd --zone=public --list-all #查看public区域规则

虚拟机desktop

]# firefox 172.25.0.11 #访问成功

]# firefox ftp://172.25.0.11 #访问成功

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

#### 永久在public区域中添加协议

• 在配置规则的位置永久(--permanent)

以public区域为例，其他区域也可添加或删除协议

]# firewall-cmd --reload #重新加载防火墙所有规则

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

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

]# firewall-cmd --zone=public --list-all #查看public区域规则

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

]# firewall-cmd --reload #重新加载防火墙所有规则

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

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

#### 例如单独拒绝虚拟机desktop

将虚拟机desktop的IP地址放入block区域中，在虚拟机server:设置防火墙规则

]# firewall-cmd --permanent --zone=block --add-source=172.25.0.10

]# firewall-cmd --reload

]# firewall-cmd --zone=block --list-all #查看

测试:

[root@desktop0 ~]# firefox 172.25.0.11 #刷新浏览器访问,失败

[root@desktop0 ~]# ping 172.25.0.11 #失败

[root@room9pc01 ~]# firefox 172.25.0.11 #真机访问成功

#### 删除规则:

[root@server0 ~]# firewall-cmd --permanent --zone=block --remove-source=172.25.0.10

[root@server0 ~]# firewall-cmd --reload

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

### 端口

端口: 协议或服务或程序编号起到标识作用

root可以改变端口,一个程序具备多个端口

IP地址作用:在互联网寻找到相应的主机

端口作用:在主机标识，协议或服务或程序

数据包:源IP地址 目标IP地址 数据 目标端口

#### 互联网常见的协议及端口:

http:超文本传输协议 默认端口: 80

https:安全的超文本传输协议 默认端口: 443

ftp:文本传输协议 默认端口:21

tftp:简单的文本传输协议 默认端口:69

DNS:域名解析协议 默认端口:53

telnet:远程管理协议 默认端口:23

SMTP:邮件协议(发邮件的协议) 默认端口:25

pop3:邮件协议(收邮件的协议) 默认端口:110

SNMP:简单网络管理协议 默认端口:161

#### 实现本机的端口转发

• 本地应用的端口重定向(端口1 --> 端口2)

– 从客户机访问 5423 的请求,自动映射到本机 80

– 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

虚拟机desktop:

[root@desktop0 ~]# firefox 172.25.0.11:5423 #访问失败

虚拟机server:

]# 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desktop:

[root@desktop0 ~]# firefox 172.25.0.11:5423 #访问成功

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

# 应用管理ENGINEER04

## Samba服务

Samba服务基础:跨平台的共享 Windows与Linux

• 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协议 139端口)，主要用来用户名和密码验证

CIFS(TCP协议 445端口)，主要用来传输、存储数据

### 所需软件包:samba

### 系统服务:smb

### 管理共享账号

• 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)

### 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

常见用户管理办法：

创建一个指定/sbin/nologin解释器的用户，再将此用户添加位samba用户

]#useradd -s /sbin/nologin zdd

]#pdbedit -a zdd

New passwd: #设置zdd的samba用户密码

]#pdbedit -l zdd

]#pdbedit -x zdd

### 配置 /etc/samba/smb.conf文件

[自定共享名]

path = 文件夹绝对路径

; public = no|yes //默认no 与Windows的guest来宾用户有关

; browseable = yes|no //默认yes 与隐藏共享有关

; read only = yes|no //默认yes

; write list = 用户1 .. .. //默认无

; valid users = 用户1 .. .. //默认为所有Samba共享帐号

; hosts allow = 客户机地址 .. ..

; hosts deny = 客户机地址 .. ..

### Server服务端只读实例操作步骤

#### 1.安装服务端软件samba

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

#### 2.创建共享帐号

[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 #添加为samba共享帐号

[root@server0 ~]# pdbedit -a kenji

[root@server0 ~]# pdbedit -a chihiro

[root@server0 ~]# pdbedit -L #查看samba共享帐号

#### 3.创建共享目录

[root@server0 ~]# mkdir /common

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

[root@server0 ~]# ls /common/

#### 4.修改配置文件/etc/samba/smb.conf

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

#此服务器必须是 STAFF 工作组的一个成员

workgroup = STAFF

#命令模式下,按大写的G到全文的最后

[common] #共享名

path = /common #共享的实际路径

#### 5.重起smb服务,设置为开机自起服务

[root@server0 ~]# systemctl restart smb

[root@server0 ~]# systemctl enable smb

#### 6.修改SELinux策略,布尔值(服务功能开和关)

]# getsebool -a | grep samba #查看布尔值

]# setsebool samba\_export\_all\_ro on #修改布尔值,开启只读访问

]# getsebool -a | grep samba

需要加 -P 选项才能实现永久设置,足够内存

### desktop客户端实例操作步骤

#### 1.安装客户端软件samba-client

[root@desktop0 ~]# yum -y install samba-client

#### 2.使用客户端软件提供的命令,访问服务端共享

[root@desktop0 ~]# smbclient -L 172.25.0.11

Enter root's password:

Anonymous login successful #以匿名用户进行登陆

Domain=[STAFF] OS=[Unix] Server=[Samba 4.1.1]

Sharename Type Comment

--------- ---- -------

common #共享名为common

[root@desktop0 ~]# smbclient -U harry //172.25.0.11/common

Enter harry's password: #输入密码

Domain=[STAFF] OS=[Unix] Server=[Samba 4.1.1]

smb: \> ls

注意：防火墙、SElinux模式

### desktop客户端更加方便的访问:

#### 安装cifs-utils,

cifs-utils：支持cifs协议及相应文件系统

[root@desktop0 ~]# yum -y install cifs-utils

#### 2.进行挂载

[root@desktop0 ~]# mkdir /mnt/nsd

[root@desktop0 ~]# ls /mnt/nsd

]# mount -o user=harry,pass=123 //172.25.0.11/common /mnt/nsd/

[root@desktop0 ~]# df -h #查看samba共享挂载情况

[root@desktop0 ~]# ls /mnt/nsd

[root@desktop0 ~]# umount /mnt/nsd/

#### 3.开机自动挂载:修改配置/etc/fstab

\_netdev:声明网络设备,在挂载该设备时,需要配置本机IP地址及网络服务正常才能挂载

//172.25.0.11/common /mnt/nsd cifs defaults,user=harry,pass=123,\_netdev 0 0

[root@desktop0 ~]# df -h

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

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

### Server配置可读写Samba共享

#### 1.创建共享的目录

[root@server0 ~]# mkdir /devops

[root@server0 ~]# echo abc > /devops/a.txt

[root@server0 ~]# ls /devops/

#### 2. 修改/etc/samba/smb.conf

.......

[devops] #共享名

path = /devops #共享的实际路径

write list = chihiro #允许chihiro用户可以写入

#### 3.重起smb服务

[root@server0 ~]# systemctl restart smb

#### 4.修改SELinux策略,布尔值 开放读写权限

[root@server0 ~]# getsebool -a | grep samba

[root@server0 ~]# setsebool samba\_export\_all\_rw on

[root@server0 ~]# getsebool -a | grep samba

#### 5.设置本地目录权限

[root@server0 ~]# setfacl -m u:chihiro:rwx /devops/

[root@server0 ~]# getfacl /devops/

#### 6.客户端开机自动挂载

[root@desktop0 ~]# vim /etc/fstab

//172.25.0.11/devops /mnt/dev cifs defaults,user=chihiro,pass=123,\_netdev 0 0

[root@desktop0 ~]# mkdir /mnt/dev

[root@desktop0 ~]# mount -a

[root@desktop0 ~]# df -h

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

客户端访问服务端

1.防火墙

2.安全SELinux

3.服务本身的访问控制

4.本地目录权限

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

总结:读写Samba共享的搭建

服务端:

1.设置防火墙默认区域为trusted

2.安装软件包samba

3.常见共享用户的帐号 pdbedit

4.创建共享目录

5.修改配置文件/etc/samba/smb.conf发布共享

指明 write list = 可写用户

6.重起smb服务

7.开放SELinux读写布尔值

8.设置本地目录的权限

客户端:

1.设置防火墙默认区域为trusted

2.安装软件cifs-utils

3.挂载访问

6.实现开机自动挂载

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

了解内容:multiuser机制

• SMB客户端的 multiuser 挂载技术

– 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身份(无需重新挂载)

• mount.cifs 的挂载参数

– 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

## 配置NFS共享

配置NFS共享: Linux与Linux平台之间的共享

普通NFS服务：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

### server服务端

#### 1．安装nfs-utils软件

[root@server0 ~]# yum -y install nfs-utils

#### 2．创建共享目录

[root@server0 ~]# mkdir /nfs

[root@server0 ~]# echo haha > /nfs/1.txt

[root@server0 ~]# ls /nfs

#### 3．修改配置文件/etc/exports 发布共享

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

– 文件夹路径 客户机地址(权限)

/nfs \*(ro)

#### 4．重起服务nfs-server

[root@server0 ~]# systemctl restart nfs-server

[root@server0 ~]# systemctl enable nfs-server

### Desktop客户端虚拟机desktop:

**客户端要安装:nfs**

]# yum -y install nfs-utils

]# systemctl start nfs-utils

]# systemctl enable nfs-utils

#### 完成开机自动挂载/etc/fstab

172.25.0.11:/nfs /mnt/nfsmount nfs defaults,\_netdev 0 0

#### 2.进行mount -a检测

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

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

## 配置IPv6地址

### • IPv4 地址表示

– 32个二进制位,点分隔的十进制数

– 例如:172.25.0.11、127.0.0.1

### • IPv6 地址表示

– 128个二进制位,冒号分隔 的十六进制数

– 每段内连续的前置 0 可省略、连续的多个 : 可简化为 ::

– 例如: 2003:ac18:0000:0000:0000:0000:0000:0305

2003:ac18::305

[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 ~]# ping6 2003:ac18::305

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

# 应用管理ENGINEER05

## iSCSI网络磁盘

iSCSI磁盘的工作模式 (默认端口号: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

• 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命名规范)

iqn.yyyy-mm.倒序域名:自定义标识，用来识别target 磁盘组,也用来识别客户机身份

### 服务端配置

#### 一 划分一个主分区5G

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

[root@server0 ~]# lsblk

[root@server0 ~]# ls /dev/vdb1

#### 二 安装软件targetcli

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

#### 三 配置共享存储

[root@server0 ~]# targetcli

/> ls

1.创建后端存储,为其设置名字为nsd

/> backstores/block create dev=/dev/vdb1 name=nsd

/> ls

2.创建target磁盘组

/> iscsi/ create iqn.2016-02.com.example:server0

/> ls

3.Lun关联

/> iscsi/iqn.2016-02.com.example:server0/tpg1/luns create /backstores/block/nsd

/> ls

4. 配置ACL访问控制 (设置客户端访问时,声称的名字,该名字必须符合iqn命名规范)

/> iscsi/iqn.2016-02.com.example:server0/tpg1/acls create iqn.2018-12.com.example:abc

/> ls

5.启用网络接口及端口号

/> iscsi/iqn.2016-02.com.example:server0/tpg1/portals

create 172.25.0.11

/> exit

#### 四 重起服务target

[root@server0 ~]# systemctl restart target

[root@server0 ~]# systemctl enable target

### 客户端配置

#### 1.安装软件包 iscsi-initiator-utils

Yum仓库具备补全仓库包名功能

1.Yum要生成缓存

2.当前系统没有安装该软件包

[root@desktop0 ~]# rpm -q iscsi-initiator-utils

iscsi-initiator-utils-6.2.0.873-21.el7.x86\_64

#### 2.修改配置文件,指定客户端声称的名字

[root@desktop0 ~]# vim /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2018-12.com.example:abc

#### 3.重起iscsid服务,刷新客户端声称的名字

[root@desktop0 ~]# systemctl restart iscsid

Warning: Unit file of iscsid.service changed on

disk, 'systemctl daemon-reload' recommended.

[root@desktop0 ~]# systemctl daemon-reload

[root@desktop0 ~]# systemctl restart iscsid

#### 4.发现服务端共享存储

Linux查看命令帮助的方法 : man iscsiadm

全文查找/example

Ctrl - ：减小字体

Ctrl Shift + ：变大字体

[root@desktop0 ~]# iscsiadm --mode discoverydb --type sendtargets --portal 172.25.0.11 --discover

172.25.0.11:3260,1 iqn.2016-02.com.example:server0

[root@desktop0 ~]#

#### 5.重起客户端服务,识别加载共享存储

[root@desktop0 ~]# lsblk

[root@desktop0 ~]# systemctl restart iscsi

[root@desktop0 ~]# systemctl enable iscsi

[root@desktop0 ~]# lsblk

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

总结:

服务端:

1.设置防火墙默认区域为trusted

2.安装软件包targetcli

3.运行targetcli命令执行配置

a:创建后端存储

b:创建磁盘组

c:进行关联lun

d:配置ACL访问控制,客户端声称的名字

e:启用网络接口,启用端口

4.重起服务target

客户端:

1.设置防火墙默认区域为trusted

2.安装软件包 iscsi-initiator-utils

3.修改配置文件,指定客户端声称的名字

4.重起iscsid刷新客户端声称的名字

5.执行命令发现服务端共享存储

6.重起客户端服务iscsi,识别加载共享存储

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

## 数据库服务基础

什么是数据库:存放数据的仓库

数据库中有很多的库,在每一个库中,会有很多的表格

• 常见的关系型 数据库管理系统

– 微软的 SQL Server

– IBM的 DB2

– 甲骨文的 Oracle、MySQL

– 社区开源版 MariaDB

### 一 部署mariadb数据库

mariadb服务器(默认端口:3306)

• RHEL7 中的 MariaDB 相关包

– mariadb-server:提供服务端有关的系统程序

– mariadb:提供客户端及管理工具

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

[root@server0 ~]# systemctl restart mariadb

[root@server0 ~]# systemctl enable mariadb

### 二 数据库的基本操作

[root@server0 ~]# mysql

MariaDB [(none)]> show databases; #显示所有的库

MariaDB [(none)]> create database nsd; #创建nsd库

MariaDB [(none)]> show databases;

MariaDB [(none)]> drop database nsd; #删除nsd库

MariaDB [(none)]> create database nsd1811; #创建nsd1811库

MariaDB [(none)]> show databases;

MariaDB [(none)]> exit

### 三 为数据库账号修改密码

数据库管理员:账户名 root 信息放在mysql库中,user表

系统管理员:账户名 root 信息放在/etc/passwd

– 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 #交互式登陆

Enter password:

[root@server0 ~]# mysql -u root -p123 #非交互式登陆

### 四 禁止监听网络,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

.. ..

[root@server0 ~]# systemctl restart mariadb

### 五 导入/恢复到数据库

– mysql [-u用户名] [-p[密码]] 数据库名 < 备份文件.sql

[root@server0 ~]# wget http://classroom.example.com/pub/materials/users.sql

[root@server0 ~]# ls

[root@server0 ~]# mysql -u root -p123 nsd1811 < users.sql

[root@server0 ~]# mysql -u root -p123

MariaDB [(none)]> show databases;

MariaDB [(none)]> use nsd1811; #进入nsd1811库

MariaDB [nsd1811]>show tables; #查看库中所有表格

表格: 表记录 表字段

增(insert) 删(delete) 改(update) 查(select)

### 查(select)

查询格式: select 表字段 from 库名.表名;

[root@server0 ~]# mysql -u root -p123

MariaDB [nsd1811]> use nsd1811; #进入nsd1811库

MariaDB [nsd1811]> show tables; #查看当前所有库表格

MariaDB [nsd1811]> select \* from base; #查看所有表字段信息

MariaDB [nsd1811]> select \* from location;

MariaDB [nsd1811]> use mysql; #进入mysql库

MariaDB [mysql]> show tables;

MariaDB [mysql]> select \* from nsd1811.base;

#### 有条件的查询：where

有条件的查询: select 表字段 from 库名.表名 where 表字段='值';

> use nsd1811;

> select \* from base where password='123';

> select \* from base where name='james';

> select id,name from base where password='456';

#### 查看表结构: desc 表名;

> use mysql;

> show tables;

> desc user;

> select user,host,password from user;

### 用户授权设置

– 除了root用户,此nsd1811库只能被用户 lisi查询,此用户的密码为123

授权命令:

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';

grant select on nsd1811.\* to lisi@localhost identified by '123';

[root@server0 ~]# mysql -u root -p123

> grant select on nsd1811.\* to lisi@localhost identified by '123';

#当lisi从本地localhost登陆后,输入密码123将会获得nsd1811库中所有表的查询权限

> select user,host,password from mysql.user;

> exit

[root@server0 ~]# mysql -u lisi -p123 #利用lisi登陆测试

MariaDB [(none)]> select \* from nsd1811.base;

MariaDB [(none)]> exit

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

### 数据库查询案例

在系统 server0 上使用数据库 nsd1811,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 nsd1811;

> select \* from base where password='solicitous';

> select \* from base;

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;

### 增(insert)

> insert base values ('6','Barbara','111222');

> select \* from base;

> insert location values ('6','Sunnyvale');

> select \* from location;

### 删(delete)

1. 禁止空密码root用户访问 mariadb 数据库

> use mysql;

> select user,host,password from user;

> select user,host,password from user where password='';

> delete from user where password=''; #删除密码为空的记录

> select user,host,password from user;

> flush privileges; #刷新user表记录

> exit

[root@server0 ~]# mysql -u root -h server0.example.com #-h:登陆方式

#登陆失败

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

# 应用管理ENGINEER06

## HTTP服务基础（Apache）：

* http：基于B/S（Browser/server）架构的网页服务
* HTML协议：超文本标记语言 HTTP协议：超文本传输协议 默认端口：80
* Web网站服务端：软件包httpd、系统服务httpd
* Web网站浏览器：软件包elinks或fireox
* 传输协议及端口：TCP 80
* Web网站服务端主配置文件：/etc/httpd/conf/httpd.conf
* Web网站服务端调用配置文件：/etc/httpd/conf.d/\*.conf
* 默认首页文件：index.html
* httpd网站文档的默认根目录:/var/www/html
* URL（Uniform Resource Locator，统一资源定位器）网址的基本组成：
* http://服务器地址[:端口号]/目录/文件名
* 对于需要验证的FTP资源，还需要指定用户名密码信息：
* ftp://用户名:密码@服务器地址[:端口号]/目录/文件名
* 安装了httpd之后，系统自动创建apache用户，由此用户读取网页文件

## 1 案例1：独立Web站点的快速部署

1.1 问题

本例要求为 http://server0.example.com 配置Web站点，要求如下：

从http://classroom/pub/materials/station.html下载一个主页文件，将其重命名为 index.html将此文件拷贝到站点的 DocumentRoot 目录下，不要对文件 index.html 的内容作任何修改使用 elinks 或firefox 浏览上述Web站点

### 步骤一：构建及部署网站服务器

1）安装软件包httpd

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

.. ..

2）部署网页

[root@server0 ~]# cd /var/www/html/ //进入网页目录

[root@server0 html]# wget http://classroom/pub/materials/station.html -O index.html //下载网页

.. ..

2016-11-26 19:33:49 (1.36 MB/s) - ‘index.html’ saved [14/14]

[root@server0 html]# cat index.html //检查网页文件

Default Site.

3）配置主配置文件：/etc/httpd/conf/httpd.conf

[root@server0 html]#vim /etc/httpd/conf/httpd.conf

主配置文件中定义了用来调用/etc/httpd/conf.d/ 目录下新建的\*.conf 配置文件

\*.conf文件需自己建立，并配置

4）启动系统服务httpd，并设置开机自启

[root@server0 html]# systemctl restart httpd

[root@server0 html]# systemctl enable httpd

### 步骤二：访问网站服务器

1）使用elinks浏览器查看

Elinks浏览器可以在命令行模式显示出网页文本，经常用来测试网站的可用性。

[root@desktop0 ~]# yum -y install elinks //安装elinks

.. ..

[root@desktop0 ~]# elinks -dump http://server0.example.com/ //访问指定网址

Default Site.

2）使用firefox浏览器查看

Firefox浏览器支持更多网页特性，是访问复杂网页、网址的优秀工具。

在桌面终端直接运行“firefox http://server0.examle.com/”，或者通过菜单快捷方式打开Firefox浏览器再输入对应网址，都可以看到目标网页（如图-1所示）。

## 2 案例2：虚拟Web主机的部署

2.1 问题

本例要求为server0扩展Web站点，新建虚拟主机 http://www0.example.com，要求如下：

设置 DocumentRoot 为 /var/www/virtual。从 http://classroom/pub/materials/www.html 下载主页文件，并重命名为 index.html。不要对文件 index.html 的内容作修改，将其放到此虚拟主机的 DocumentRoot 目录下。确保 fleyd 用户能在 /var/www/virtual 目录建文件

确保站点 http://server0.example.com 仍然可用

2.2 方案

单一网站平台（比如172.25.0.11）：

多个域名 ---> 相同的网页内容

主配置文件：/etc/httpd/conf/httpd.conf

网页目录定义：DocumentRoot /var/www/html

虚拟主机平台（比如172.25.0.11）：

创建调用配置文件：/etc/httpd/conf.d/\*.conf #自主创建

在同一套httpd平台上跑很多个网站，多个域名 ---> 不同的网页内容

网页目录由<VirtualHost ...>区段配置定义

多个虚拟主机站点的典型设置（/etc/httpd/conf.d/\*.conf）：

<VirtualHost \*:80>

ServerName server0.example.com #(网站1的FQDN)

DocumentRoot /var/www/html 网站1的网页根目录

</VirtualHost>

<VirtualHost \*:80>

ServerName www0.example.com #(网站2的FQDN)

DocumentRoot /var/www/virtual 网站2的网页根目录

</VirtualHost>

.. ..(可再添加)

FQDN：全限定域名：同时带有主机名和域名的名称，例如：www.example.com

部署了虚拟web主机，所有的web站点都必须使用虚拟web主机功能实现

### 步骤一：部署网页文档

1）建立网页目录

[root@server0 ~]# mkdir /var/www/virtual

[root@server0 ~]# useradd fleyd

[root@server0 ~]# setfacl -m u:fleyd:rwx /var/www/virtual/

2）部署网页文件

[root@server0 ~]# cd /var/www/virtual/

[root@server0 virtual]# wget http://classroom/pub/materials/www.html -O index.html

.. ..

100%[=====================>] 14 --.-K/s in 0s

2016-11-26 20:01:14 (826 KB/s) - ‘index.html’ saved [14/14]

[root@server0 virtual]# cat index.html //检查网页文件

Virtual Site.

### 步骤二：配置虚拟主机 http://www0.example.com/

1）为新站点创建独立的配置文件

[root@server0 virtual]# vim /etc/httpd/conf.d/01-www0.conf

<VirtualHost \*:80>

ServerName www0.example.com

DocumentRoot /var/www/virtual

</VirtualHost>

[root@server0 virtual]# httpd -t //确保语法检查OK

Syntax OK

2）重启系统服务httpd

[root@server0 virtual]# systemctl restart httpd

### 步骤三：访问虚拟主机 http://www0.example.com/

访问此虚拟站点，可以看到预期的网页内容：

[root@desktop0 ~]# elinks -dump http://www0.example.com/

Virtual Site.

### 步骤四：完善原始站点 http://server0.example.com/

需要注意的是，原始的独立站点可能出现异常，访问时并不是原始的网页：

[root@desktop0 ~]# elinks -dump http://server0.example.com/

Virtual Site.

原因是一旦启用虚拟站点机制以后： 外部的 DocumentRoot、ServerName 会被忽略

第1个虚拟站点被视为默认站点，若客户机请求的URL不属于任何已知站点，则由第1个站点响应

若要解决此异常，需要将原始站点转换为第一个虚拟主机，启用顺序的设置可以通过文件名开头的数字来实现。

1）为原始站点建立虚拟主机配置

[root@server0 ~]# vim /etc/httpd/conf.d/00-default.conf

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /var/www/html

</VirtualHost>

2）重启系统服务httpd

[root@server0 virtual]# systemctl restart httpd

3）访问两个虚拟站点，确保各自的网页内容正确

[root@desktop0 ~]# elinks -dump http://server0.example.com/

Default Site.

[root@desktop0 ~]# elinks -dump http://www0.example.com/

Virtual Site.

## 3 案例3：配置网页内容访问（访问控制）

3.1 问题

本例要求在 Web 网站 http://server0.example.com 的 DocumentRoot 目录下创建一个名为 private 的子目录，要求如下：

从 http://classroom/pub/materials/private.html 下载一个文件副本到这个目录，重命名为 index.html

不要对文件 index.html 的内容作任何修改

从 server0 上，任何人都可以浏览 private 的内容，但是从其他系统不能访问这个目录的内容

3.2 方案

配置Web内容的访问控制需要添加Directory区段，主要形式可参考

<Directory "父目录路径">

Require all denied //上层目录拒绝任何访问

</Directory>

<Directory "子目录1路径">

Require all granted //子目录1允许任何访问

</Directory>

<Directory "子目录2路径">

Require ip IP或网段地址 .. .. //子目录2允许少数客户机

</Directory>

### 步骤一：部署网页子目录及文档

1）建立子目录

[root@server0 ~]# mkdir /var/www/html/private

2）部署网页

[root@server0 ~]# cd /var/www/html/private

[root@server0 private]# wget http://classroom/pub/materials/private.html -O index.html

.. ..

2016-11-26 20:30:28 (1.90 MB/s) - ‘index.html’ saved [14/14]

[root@server0 private]# cat index.html //检查网页文件

Private Site.

### 步骤二：为指定的网页子目录限制访问

在httpd服务的标准配置中，根目录 / 默认拒绝任何访问，但网页目录/var/www/默认允许任何访问。因此，只需要为个别子目录增加访问控制即可。

1）调整虚拟站点server0.example.com的配置文件

[root@server0 ~]# vim /etc/httpd/conf.d/00-default.conf

.. ..

<Directory "/var/www/html/private">

Require ip 127.0.0.1 ::1 172.25.0.11

</Directory>

2）重启系统服务httpd

[root@server0 ~]# systemctl restart httpd

### 步骤三：测试目录访问限制

1）从desktop0上访问http://server0.example.com/private/被拒绝

[root@desktop0 ~]# elinks -dump http://server0.example.com/private/

Forbidden

You don't have permission to access /private/ on this server.

2）从desktop0上访问http://server0.example.com/仍然是正常的

[root@desktop0 ~]# elinks -dump http://server0.example.com/

Default Site.

3）从server0本机上访问http://server0.example.com/private/也不受限制

[root@server0 ~]# elinks -dump http://server0.example.com/private/

Private Site.

## 4 案例4：使用自定Web根目录

4.1 问题

本例要求调整 Web 站点 http://server0.example.com 的网页目录，要求如下：

新建目录 /webroot，作为此站点新的网页目录

从 http://classroom/pub/materials/station.html 下载一个文件副本到这个目录，重命名为 index.html

不要对文件 index.html 的内容作任何修改

确保站点 http://server0.example.com 仍然可访问

4.2 方案

在SELinux强制启用模式下，增加新的合规网页目录的方法：

1）参照标准目录，重设新目录的属性

chcon [-R] --reference=模板目录 新目录 或者

2）将新目录增加到预设的标准Web目录范围

semanage fcontext -a -t httpd\_sys\_content\_t '新目录(/.\*)?'

### 步骤一：部署网页目录及文档

1）建立网页目录

[root@server0 ~]# mkdir /webroot

2）部署网页文件

[root@server0 ~]# cd /webroot/

[root@server0 webroot]# wget http://classroom/pub/materials/station.html -O index.html

.. ..

2016-11-26 20:01:14 (826 KB/s) - ‘index.html’ saved [14/14]

[root@server0 webroot]# cat index.html //检查网页文件

Default Site.

### 步骤二：调整虚拟站点http://server0.example.com/的配置

1）修改配置文件

[root@server0 ~]# vim /etc/httpd/conf.d/00-default.conf

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /webroot

</VirtualHost>

.. ..

2）重启系统服务httpd

[root@server0 ~]# systemctl restart httpd

### 步骤三：确保虚拟站点http://server0.example.com/仍然可以访问

1）未调整网页目录SELinux上下文件的情况(SElinux 安全机制)

为虚拟站点http://server0.example.com/更换了新的网页目录以后，从浏览器访问将会失败，只能看到红帽测试页。

[root@desktop0 ~]# elinks -dump http://server0.example.com/

Red Hat Enterprise Linux Test Page

This page is used to test the proper operation of the Apache HTTP server

after it has been installed. If you can read this page, it means that the

Apache HTTP server installed at this site is working properly.

.. ..

针对此问题，可以参考目录/var/www的属性为网页目录/webroot设置SELinux安全上下文。

[root@server0 ~]# chcon -R --reference=/var/www /webroot/

[root@server0 ~]# ls -Z /webroot/index.html //确认结果

-rw-r--r--. root root system\_u:object\_r:httpd\_sys\_content\_t:s0 /webroot/index.html

2）未配置目录内容访问的情况

尽管已经调整过/webroot的SELinux安全上下文，但是从浏览器访问此虚拟站点时仍然会被拒绝，还是只能看到红帽测试页。

还需要修改对应的配置文件，添加内容访问控制：

[root@server0 ~]# vim /etc/httpd/conf.d/00-default.conf

<VirtualHost \*:80>

ServerName server0.example.com

DocumentRoot /webroot

</VirtualHost>

<Directory "/webroot">

Require all granted

</Directory>

<Directory "/webroot/private">

Require ip 127.0.0.1 ::1 172.25.0.11

</Directory>

[root@server0 ~]# systemctl restart httpd //重启httpd服务

若要保持原有private子目录，建议也拷贝过来：

[root@server0 ~]# cp -rf /var/www/html/private/ /webroot/

3）最终访问测试

从浏览器能成功访问调整后的虚拟站点http://server0.example.com/。

[root@desktop0 ~]# elinks -dump http://server0.example.com/

Default Site.

## 5案例5：配置Alias实现网页跳转

当客户端访问一个网络路径页面，跳转到实际服务器本地路径

例如： webapp0.example.com 配置的根目录 DocumentRoot为 /var/www/webapp0，访问webapp0.example.com时应该返回的是/var/www/webapp0目录里面的网页文件，而Alias可以将这个返回页面跳转到另外指定的目录里面的页面，如下跳转到/var/www/webapp0/webinfo.wsgi

<VirtualHost \*:80>

DocumentRoot /var/www/webapp0

ServerName webapp0.example.com

Alias / /var/www/webapp0/webinfo.wsgi

</VirtualHost>

配置语句Alias / /var/www/webapp0/webinfo.wsgi 的作用是：当访问这个网站的根目录里面的默认网页时，将会跳转到/var/www/webapp0/webinfo.wsgi网页

## 6案例6：部署并测试WSGI站点，且基于端口

5.1 问题

本例要求为站点 webapp0.example.com 配置提供动态Web内容，要求如下：

此虚拟主机侦听在端口8909

测试网页从以下地址下载，不要作任何更改http://classroom/pub/materials/webinfo.wsgi

从浏览器访问 http://webapp0.example.com:8909 可接收到动态生成的 Web 页面

此站点必须能被 example.com 域内的所有系统访问

5.2 方案

为httpd增加对Python网页程序的支持，可以安装mod\_wsgi模块。关于此模块的配置说明，建议参考软件包提供的readme文档。

在SELinux处于Enforcing模式时，若要开放非80、81等常规Web端口，需要调整SELinux保护策略。

### 步骤一：部署动态网页文档

1）创建网页目录

[root@server0 ~]# mkdir /var/www/webapp0

2）部署webinfo.wsgi网页程序

[root@server0 ~]# cd /var/www/webapp0

[root@server0 webapp0]# wget http://classroom/pub/materials/webinfo.wsgi

.. ..

2016-11-27 01:52:26 (16.0 MB/s) - ‘webinfo.wsgi’ saved [397/397]

[root@server0 webapp0]# cat webinfo.wsgi //检查下载文件

#!/usr/bin/env python

import time

.. ..

### 步骤二：配置新的虚拟主机http://webapp0.example.com：8909/

1）安装mod\_wsgi模块软件包

[root@server0 ~]# yum -y install mod\_wsgi

.. ..

2）为新虚拟主机建立配置

[root@server0 ~]# vim /etc/httpd/conf.d/02-webapp0.conf

Listen 8909 #添加此句监听端口8909

<VirtualHost \*:8909>

DocumentRoot /var/www/webapp0

ServerName webapp0.example.com

WSGIScriptAlias / /var/www/webapp0/webinfo.wsgi

</VirtualHost>

3）调整SELinux策略，允许Web服务使用8909端口

列出当前许可的Web端口：

[root@server0 ~]# semanage port -l | grep ^http\_port

http\_port\_t tcp 80, 81, 443, 488, 8008, 8009, 8443, 9000

添加新的Web端口：

[root@server0 ~]# semanage port -a -t http\_port\_t -p tcp 8909

[root@server0 ~]#

确认配置结果：

[root@server0 ~]# semanage port -l | grep ^http\_port

http\_port\_t tcp 8909, 80, 81, 443, 488, 8008, 8009, 8443, 9000

4）重启系统服务httpd

[root@server0 ~]# systemctl restart httpd

[root@server0 ~]# netstat -antpu | grep httpd //确认已监听8909端口

tcp6 0 0 :::443 :::\* LISTEN 2477/httpd

tcp6 0 0 :::8909 :::\* LISTEN 2477/httpd

tcp6 0 0 :::80 :::\* LISTEN 2477/httpd

### 步骤三：测试动态网页效果

使用elinks或firefox访问此动态站点http://webapp0.example.com:8909/。

多刷新访问几次，每次看到的是动态网页内容，内容并不固定。

[root@desktop0 ~]# elinks -dump http://webapp0.example.com:8909/

UNIX EPOCH time is now: 1480184916.52 //第1次访问

[root@desktop0 ~]# elinks -dump http://webapp0.example.com:8909/

UNIX EPOCH time is now: 1480184919.21 //第2次访问

[root@desktop0 ~]# elinks -dump http://webapp0.example.com:8909/

UNIX EPOCH time is now: 1480184951.99 //第3次访问

6案例6：

## 扩展1：SElinux 安全上下文（标签）

复制目录属性：chcon [-R] --reference=模板目录 新目录

[root@server0 ~]# mkdir /webroot

[root@server0 ~]# ls –Zd /var/www/ #可看见目录有httpd的标签

[root@server0 ~]# ls –ZS /webroot #目录没有httpd 的标签

[root@server0 ~]# chcon –R --reference=/var/www /webroot

[root@server0 ~]# ls –ZS /webroot #目录有了httpd 的标签，与www/目录属性相同

## 扩展2：调整SELinux策略添加查看端口

列出当前许可的Web端口：

[root@server0 ~]# semanage port -l | grep ^http\_port

http\_port\_t tcp 80, 81, 443, 488, 8008, 8009, 8443, 9000

添加新的Web端口：

[root@server0 ~]# semanage port -a -t http\_port\_t -p tcp 8909

确认配置结果：

[root@server0 ~]# semanage port -l | grep ^http\_port

http\_port\_t tcp 8909, 80, 81, 443, 488, 8008, 8009, 8443, 9000

# 应用管理ENGINEER07

## 安全Web服务

Web服务：端口的优先级,优于域名,也优于IP地址

https:安全的超文本传输协议 端口443

– 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:负责证书申请/审核/颁发/鉴定/撤销等管理工作

TLS证书部署位置：/etc/pki/tls/certs/\*.crt

TLS私钥部署位置：/etc/pki/tls/private/\*.key

1.安装httpd软件

2.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

3.部署根证书(工商局信息)

]# cd /etc/pki/tls/certs/

]# wget http://classroom.example.com/pub/example-ca.crt

4.部署解密的私钥

]# cd /etc/pki/tls/private/

]# wget http://classroom.example.com/pub/tls/private/server0.key

5.安装mod\_ssl软件,支持安全加密的Web软件

]# yum -y install mod\_ssl

]# rpm -ql mod\_ssl

/etc/httpd/conf.d/ssl.conf

6.修改配置文件

[root@server0 /]# vim /etc/httpd/conf.d/ssl.conf

在末行模式 :set nu #开启行号功能

59 DocumentRoot "/var/www/html"

60 ServerName www0.example.com:443

#指定网站证书

100 SSLCertificateFile /etc/pki/tls/certs/server0.crt

#指定解密的私钥匙文件

107 SSLCertificateKeyFile /etc/pki/tls/private/server0.key

#指定根证书(工商局信息)

122 SSLCACertificateFile /etc/pki/tls/certs/example-ca.crt

7.重起httpd服务

]# systemctl restart httpd

]# echo '<h1>NSD1811 Web' > /var/www/html/index.html

]# cat /var/www/html/index.html

8.测试验证

[root@server0 /]# firefox https://www0.example.com

## postfix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 —— 传递给收件服务器

– 处理用户收到的邮件 —— 投递到邮箱

SMTP:用户发邮件协议 默认端口 25

pop3:用户收邮件协议 默认端口 110

DNS服务器:虚拟机classroom

### 1.安装软件包postfix

[root@server0 /]# rpm -q postfix

postfix-2.10.1-6.el7.x86\_64

### 2.修改主配置文件 /etc/postfix/main.cf

[root@server0 /]# vim /etc/postfix/main.cf #末行模式下开启行号 :set nu

99 myorigin = server0.example.com #收件人与发件人补全的域名后缀

116 inet\_interfaces = all #本机所有网络接口都提供邮件功能

164 mydestination = server0.example.com

#判定收件人域名后缀server0.example.com为本域邮件

destination:目标

### 3.重起服务postfix

[root@server0 /]# systemctl restart postfix

[root@server0 /]# useradd yg

[root@server0 /]# useradd xln

#### mail命令发信操作

– mail -s '邮件标题' -r 发件人 收件人

[root@server0 /]# mail -s 'test01' -r yg xln

hahaxixihehelele

.

EOT

#### mail命令收信操作

– mail [-u 用户名]

[root@server0 /]# mail -u xln

输入邮件的编号 进行查看邮件内容

[root@server0 /]# echo BBB | mail -s 'test01' -r yg xln

[root@server0 /]# mail -u xln

### 非交互式发送邮件：

]#echo 邮件内容 | mail -s ‘邮件标题’ -r 发件人 收件人

## parted分区工具

• 使用fdisk操作>2.2TB的磁盘时

– 超出容量的磁盘将会无法识别,导致分区失效

– 如何处理大容量的磁盘?

### GPT分区模式:

GPT分区模式：最多可以划分128个主分区

最大空间支持到18EB。容量：1EB=1000PB 1PB=1000TB 1TB=1000GB

[root@server0 /]# parted /dev/vdb #对/dev/vdb磁盘设备进行分区

(parted) mktable gpt #指定分区模式

(parted) print #输出分区表查看

(parted) mkpart #划分新的分区

分区名称？ []? haha #指定分区的名称,随意写

文件系统类型？ [ext2]? ext4 #分区文件系统类型,不起实际作用，已最后格式化为准

起始点？ 0

结束点？ 2G

忽略/Ignore/放弃/Cancel? Ignore #忽略 硬盘存储分区表空间

(parted) print

(parted) unit GB #使用单位为GB

(parted) mkpart #再次划分

分区名称？ []? haha

文件系统类型？ [ext2]? ext4

起始点？ 2G

结束点？ 4G

(parted) print

## 交换分区

• 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

### 1.格式化交换文件系统

[root@server0 /]# mkswap /dev/vdb1

[root@server0 /]# blkid /dev/vdb1

[root@server0 /]# mkswap /dev/vdb2

[root@server0 /]# blkid /dev/vdb2

### 2.启用交换分区

[root@server0 /]# swapon /dev/vdb1

[root@server0 /]# swapon -s #查看组成交换空间成员信息

[root@server0 /]# swapon /dev/vdb2

[root@server0 /]# swapon -s

### 3.停用交换分区

[root@server0 /]# swapoff /dev/vdb1

[root@server0 /]# swapon -s

[root@server0 /]# swapoff /dev/vdb2

[root@server0 /]# swapon -s

### 4.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

]# swapon -a #专用于检测交换分区是否书写正确的指令

]# swapon -s #查看组成交换空间成员信息

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

## 网卡配置聚合连接

网卡聚合连接：(链路聚合、网卡组队、网卡绑定)：采用虚拟网卡设计,对网卡设备的备份

eth1 eth2 虚拟:team0

进行网卡热备份(activebackup)连接冗余

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

### 1.创建虚拟网卡team0

]# nmcli connection add type team ifname team0 con-name team0 autoconnect yes config '{"runner": {"name": "activebackup"}}'

#最后一句命令参考: man teamd.conf全文搜索/example，找到再复制粘贴

语句解释：

nmcli connection add(添加) type(类型) team(组队) ifname(网卡名) team0 con-name(配置文件名) autoconnect yes (每次开机自动启用) config(配置运行模式) (热备份模式)

# ifconfig #查看team0网卡信息

# cat /et(tab)/sysco(tab)/netw(tab)-(tab)/ifcfg-team0 #查看生成的网卡配置文件信息

如果有敲错，一定要删除 : ]#nmcli connection delete team0

### 2. 添加成员eth1、eth2

]#nmcli connection add type team-slave ifname eth1 con-name team0-1 master team0

]# nmcli connection add type team-slave ifname eth2 con-name team0-2 master team0

解析： nmcli connection add(添加) type(类型为) team-slave(team的成员)

ifname( 配置文件名) team0-1 网卡为 eth\* 主设备为 team0

如果有敲错，一定要删除 : nmcli connection delete team0-1

如果有敲错，一定要删除 : nmcli connection delete team0-2

### 3.配置team0的IP地址

[root@server0 /]# nmcli connection modify team0 ipv4.method manual ipv4.addresses 192.168.1.1/24 connection.autoconnect yes

[root@server0 /]# nmcli connection up team0 #激活配置

[root@server0 /]# ifconfig

### 4.删除指令 :

nmcli connection delete team0

nmcli connection delete team0-1

nmcli connection delete team0-2

### 5.查看team0信息的指令

[root@server0 /]# teamdctl team0 state

[root@server0 /]# ifconfig eth2 down

[root@server0 /]# teamdctl team0 state

[root@server0 /]# ifconfig eth1 down

[root@server0 /]# teamdctl team0 state

[root@server0 /]# ifconfig eth1 up

[root@server0 /]# ifconfig eth2 up

[root@server0 /]# teamdctl team0 state

# 系统与服务管理进阶SERVICES01

案例1：补充应用技巧

案例2：软连接与硬连接

案例3：man手册、zip备份

案例4：自定义yum软件仓库

案例5：发布及测试yum仓库

案例6：vim效率操作

案例7：编译安装软件包

案例8：使用systemctl工具

## 1 案例1：补充应用技巧

1.1 问题

本例要求掌握在运维中比较常用的一些扩展命令技巧的使用，完成下列小技巧操作：

采用数值形式将目录/root的权限调整为 rwx------

将记录的历史命令条数更改为 200 条

统计 /boot、/etc/pki 目录占用的空间大小

以格式“yyyy-mm-dd HH:MM” 显示系统时间

1.2 方案

date日期时间工具：

显示日期时间：date、date +"%Y-%m-%d %H:%M:%S"

调整日期时间：date -s "yyyy-mm-dd HH:MM:SS"

恢复为硬件时间：hwclock –s

### 认识linux的目录结构

认识Linux的目录层次:

– man hier

• 常见一级目录的用途

/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 存放内存映射数据，不占磁盘空间

/tmp 存放临时文件

### 权限数值化

基本权限： r=4 w=2 x=1

附加权限：sticky Bit=1 SGID=2 SUID=4

7=4+2+1 拥有r w x 权限

6=4+2 拥有r w 权限

5=4+1 拥有r x 权限

3=2+1 拥有w x 权限

1）查看原来的权限

[root@svr7 ~]# ls -ld /root/

dr-xr-x---. 22 root root 4096 3月 26 14:59 /root/

2）修改为新权限

[root@svr7 ~]# chmod 700 /root/ #设置权限属主为7，属组为0，其他人为0

3）确认权限设置结果

[root@svr7 ~]# ls -ld /root/

drwx------. 22 root root 4096 3月 26 14:59 /root/

### 历史命令

1. 调整记录条数：

修改配置文件/etc/profile，找到HISTSIZE行，将此变量的值修改为200：

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

.. ..

HISTSIZE = 200

2）确认设置结果

所有用户重新登录以后即可生效：

[root@svr7 ~]# su - root

[root@svr7 ~]# echo $HISTSIZE

200

### du统计工具

1）分别统计结果：统计 /boot、/etc/pki 目录占用的空间大小

[root@svr7 ~]# du -sh /boot/ /etc/pki/

130M /boot/

1.5M /etc/pki/

2）比较du与ls查看文件大小的差异（默认块大小4096字节）：

[root@svr7 ~]# ls -lh /etc/inittab //数据大小511字节

-rw-r--r--. 1 root root 511 Sep 16 2015 /etc/inittab

[root@svr7 ~]# du -sh /etc/inittab //实际占用4KB磁盘空间

4.0K /etc/inittab

### date查看系统时间

以格式“yyyy-mm-dd HH:MM” 显示系统时间

[root@svr7 ~]# date +"%F %R"

2016-12-26 16:23

• 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" #设置时间

## 2 案例2：软连接与硬连接

软连接：指向原始文件的路径，若原始文件被删除，连接文件将失效；原始文件可以是目录；原始文件与连接文件可以在不同的分区/文件系统

硬连接：指向原始文件的i节点档案，若原始文件被删除，连接文件仍然有效；原始文件不能是目录；原始文件与连接文件必须在同一个分区/文件系统

案例要求理解软连接与硬连接的基本差异，完成下列操作：

新建文件 file1，内容为 AAAA

为 file1 建立软连接 file1-s，对比两文件内容

为 file1 建立硬连接 file1-h，对比两文件内容

对比上述 3 个文件的 i 节点编号

删除文件 file1 ，再查看文件 file1-s、file1-h 内容

### 步骤一：使用ln命令为文档/目录建立连接

1）新建一个测试文件

[root@svr7 ~]# vim file1

AAAA

2）为文件file1建立软连接file1-s并测试

[root@svr7 ~]# ln -s file1 file1-s

[root@svr7 ~]# cat file1-s

linux.tedu.cn

3）为文件file1建立硬连接file1-h并测试

[root@svr7 ~]# ln file1 file1-h

[root@svr7 ~]# cat file1-h

linux.tedu.cn

4）对比原始文件、软连接、硬连接的属性

可以发现软连接只是一个快捷方式，而硬连接与原始文件的i节点编号相同，其实对应同一块磁盘存储：

[root@svr7 ~]# ls -li /root/f0\*.txt

204645793 -rw-r--r--. 2 root root 14 Jan 6 12:14 file1-h

201628464 lrwxrwxrwx. 1 root root 12 Jan 6 12:16 file1-s -> file1

204645793 -rw-r--r--. 2 root root 14 Jan 6 12:14 file1

### 步骤二：连接原始文件删除测试

1）当原始文件被删除时，软连接将会失效，而硬连接仍然可访问文件数据

[root@svr7 ~]# rm -rf file1

[root@svr7 ~]# cat file1-s

cat: file1-s: No such file or directory

[root@svr7 ~]# cat file1-h

linux.tedu.cn

2）如果已知原始文件和硬连接的路径，当原始文件丢失时，可以快速重建

[root@svr7 ~]# ln file1-h file1

[root@svr7 ~]# ls -li /root/f0\*.txt

204645793 -rw-r--r--. 2 root root 14 Jan 6 12:14 file1-h

201628464 lrwxrwxrwx. 1 root root 12 Jan 6 12:16 file1-s -> file1

204645793 -rw-r--r--. 2 root root 14 Jan 6 12:14 file1

3）不支持为目录创建硬连接，但可以为目录建立软连接

[root@svr7 ~]# ln /etc/sysconfig/network-scripts/ /etc/network

ln: '/etc/sysconfig/network-scripts/': hard link not allowed for directory

[root@svr7 ~]# ln -s /etc/sysconfig/network-scripts/ /etc/interface

[root@svr7 ~]# ls -l /etc/interface

lrwxrwxrwx. 1 root root 31 Jan 6 12:28 /etc/interface -> /etc/sysconfig/network-scripts/

## 3 案例3：man手册、zip备份

3.1 问题

本例要求掌握man帮助手册的使用，以及zip压缩/解压缩的操作，完成下列任务：

查阅passwd命令、/etc/passwd配置文件的手册页

使用zip打包/usr/share/doc/qemu-kvm/目录

### 步骤一：使用man手册页获取帮助

1）查看passwd命令的手册页

[root@svr7 ~]# man passwd

PASSWD(1) User utilities PASSWD(1)

NAME

passwd - update user's authentication tokens

SYNOPSIS

passwd [-k] [-l] [-u [-f]] [-d] [-e] [-n mindays] [-x maxdays] [-w

warndays] [-i inactivedays] [-S] [--stdin] [username]

DESCRIPTION

The passwd utility is used to update user's authentication token(s).

This task is achieved through calls to the Linux-PAM and Libuser API.

Essentially, it initializes itself as a "passwd" service with Linux-

PAM and utilizes configured password modules to authenticate and then

update a user's password.

2）查看/etc/passwd配置文件的手册页

[root@svr7 ~]# man 2 passwd

PASSWD(5) Linux Programmer's Manual PASSWD(5)

NAME

passwd - password file

DESCRIPTION

The /etc/passwd file is a text file that describes user login

accounts for the system. It should have read permission allowed for

all users (many utilities, like ls(1) use it to map user IDs to user‐

names), but write access only for the superuser.

In the good old days there was no great problem with this general

read permission. Everybody could read the encrypted passwords, but

the hardware was too slow to crack a well-chosen password, and more‐

over the basic assumption used to be that of a friendly user-commu‐

nity. These days many people run some version of the shadow password

suite, where /etc/passwd has an 'x' character in the password

.. ..

### 步骤二：使用zip命令制作压缩包

zip/unzip压缩与解压缩：

制作zip压缩包：zip [-r] 备份文件.zip 被归档的文档...

释放zip压缩包：unzip 备份文件.zip [-d] 目标文件夹

1）将目录/usr/share/doc/qemu-kvm/备份为/root/qemu-kvm.zip

[root@svr7 ~]# zip -r /root/qemu-kvm.zip /usr/share/doc/qemu-kvm/

adding: usr/share/doc/qemu-kvm/ (stored 0%)

adding: usr/share/doc/qemu-kvm/COPYING (deflated 62%)

adding: usr/share/doc/qemu-kvm/COPYING.LIB (deflated 65%)

adding: usr/share/doc/qemu-kvm/Changelog (deflated 61%)

adding: usr/share/doc/qemu-kvm/LICENSE (deflated 45%)

adding: usr/share/doc/qemu-kvm/README (deflated 4%)

2）恢复测试

删除目标文件夹并确认结果：

[root@svr7 ~]# rm -rf /usr/share/doc/qemu-kvm/

[root@svr7 ~]# ls /usr/share/doc/qemu-kvm/

ls: cannot access /usr/share/doc/qemu-kvm/: No such file or directory

恢复目标文件夹并确认结果：

[root@svr7 ~]# unzip /root/qemu-kvm.zip -d /

Archive: /root/qemu-kvm.zip

creating: /usr/share/doc/qemu-kvm/

inflating: /usr/share/doc/qemu-kvm/COPYING

inflating: /usr/share/doc/qemu-kvm/COPYING.LIB

inflating: /usr/share/doc/qemu-kvm/Changelog

.. ..

[root@svr7 ~]# ls /usr/share/doc/qemu-kvm/

COPYING README qemu-tech.html

COPYING.LIB README.rhel6-gpxe-source qmp-commands.txt

Changelog README.systemtap qmp-events.txt

LICENSE qemu-doc.html qmp-spec.txt

## 4 案例4：自定义yum软件仓库

4.1 问题

本例要求在CentOS真机上利用RHEL7的光盘镜像文件准备一个软件仓库目录，完成下列任务： 创建目录 /var/www/html/rh7dvd

挂载 rhel-server-7.2-x86\_64-dvd.iso 到上述目录

另外，利用收集的一些第三方RPM软件包文件，配置为可发布的yum仓库目录，相关任务如下：

挂载RHEL7光盘镜像文件到 /var/www/html/ 目录

下载 LibreOffice 办公软件的 rpm 集合版文件

将其中的内容释放到 /opt/libreoffice/ 目录

为 /opt/libreoffice/ 目录建立仓库档案

4.2 方案

作为yum软件源的目录需要准备的内容：

大量的 .rpm 软件安装包文件

针对这些软件包的 repodata/ 仓库档案

repodata/ 仓库档案提供的数据：

filelists.xml.gz：提供所有软件包的文件安装清单

primary.xml.gz：提供所有软件包的基本/主要信息

other.xml.gz：提供所有软件包的其他信息

repomd.xml：提供上述档案数据文件.xml.gz的下载和校验信息

### 步骤一：准备 /var/www/html/rh7dvd 仓库目录

1）创建目录 /var/www/html/rh7dvd

[root@room9pc13 ~]# mkdir /var/www/html/rh7dvd

2）挂载 rhel-server-7.2-x86\_64-dvd.iso 到上述目录

[root@room9pc13 ~]# vim /etc/fstab

.. .

/ISO/rhel-server-7.2-x86\_64-dvd.iso /var/www/html/rh7dvd iso9660 loop,ro 0 0

[root@room9pc13 ~]# mount -a

3）确认部署结果

[root@room9pc13 ~]# ls /var/www/html/rh7dvd/

addons images Packages RPM-GPG-KEY-redhat-release

EFI isolinux release-notes TRANS.TBL

EULA LiveOS repodata

GPL media.repo RPM-GPG-KEY-redhat-beta

### 步骤二：准备 /opt/libreoffice/ 仓库目录

1）将获取的LibreOffice软件集合包释放到指定目录

[root@room9pc13 ~]# ls LibreOffice\_5.1.6.2\_Linux\_x86-64\_rpm.zip

LibreOffice\_5.1.6.2\_Linux\_x86-64\_rpm.zip

[root@room9pc13 ~]# unzip LibreOffice\_5.1\*.zip -d /opt/libreoffice

Archive: LibreOffice\_5.1.6.2\_Linux\_x86-64\_rpm.zip

inflating: /opt/libreoffice/install

creating: /opt/libreoffice/langpack\_zh-CN/

creating: /opt/libreoffice/langpack\_zh-CN/RPMS/

.. .. //释放到 /opt/libreoffice 目录

2）使用createrepo建立档案

[root@room9pc13 ~]# createrepo /opt/libreoffice/ #将会创建repodata文件

Spawning worker 0 with 53 pkgs

Workers Finished

Saving Primary metadata

Saving file lists metadata

Saving other metadata

Generating sqlite DBs

Sqlite DBs complete

3）确认repodata/档案资料

[root@room9pc13 ~]# ls /opt/libreoffice/repodata

1a5d8311268f33ad2cbf91382110e1ef9875aeea366897253a5d27fd42f9e317-other.xml.gz

2cd176f0b00492c0c13e0a659eda7dedeb1ab526dec5fd7c9bac7758558770d2-filelists.xml.gz

6ecab3585a93a917202e177b9569046238332af449a6492fcace96ea79374668-filelists.sqlite.bz2

b09a1f838262e0b67a5ab0d6be516bd921a914afd89864e09650f95662a20371-primary.sqlite.bz2

b6fcf4a24de5dc08585bf52bd34be50d7df376d5fbcf50903bfd3c1dfdf160bf-other.sqlite.bz2

f57724cf309cc102b2ee25596bf8fb39db3c23880404209ac1e379f7b9fd5c49-primary.xml.gz

repomd.xml

**安装oneko猫、sl火车、cmatrix**

1.准备好tools.tar.gz包，放到server上

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

[root@svr7 ~]# ls /

[root@svr7 ~]# ls /tools/

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

2.创建仓库数据文件(描述仓库所有包的数量及信息)

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

[root@svr7 ~]# ls /tools/other/ #多了repodata文件仓库清单

3.书写客户端配置文件

[root@svr7 ~]# vim /etc/yum.repos.d/rhel7.repo

[rhel7]

name=rhel7.4

baseurl=ftp://192.168.4.254/rhel7

enabled=1

gpgcheck=0

[other]

name=my rpm

baseurl=file:///tools/other #指定本机为Yum仓库

enabled=1

gpgcheck=0

[root@svr7 ~]# yum repolist

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

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

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

[root@svr7 ~]# sl

[root@svr7 ~]# oneko

[root@svr7 ~]# cmatrix

客户端读取的不是包在哪，而是读取的仓库数据文件repodata文件，

baseurl=\*\*\*\*\* 是指定repodata文件所在目录

## 5 案例5：发布及测试yum仓库

5.1 问题

沿用案例5，本例要求掌握发布及测试yum仓库的方法，方便在网络内提供集中的yum源服务器，主要完成下列任务：

在CentOS真机 上发布yum源，包括：rhel7 系统的光盘目录仓库、LibreOffice 的rpm软件包仓库

在主机 pc207 上使用上述yum源

5.2 方案

通过网络发布yum软件源时，只需要配置HTTP或FTP资源服务器，然后将提前准备好的yum仓库目录部署到可访问的资源位置即可。

在访问网络yum软件源时，注意客户端的baseurl地址必须与资源提供方式一致：

baseurl = http://服务器地址/目录名 ==》 /var/www/html/目录名

baseurl = ftp://服务器地址/目录位置 ==》 /var/ftp/目录名

baseurl = file:///目录位置

### 步骤一：在CentOS真机上发布yum仓库

1）快速搭建httpd服务器（若已建好，此步可跳过）

[root@room9pc13 ~]# yum -y install httpd //装包

[root@room9pc13 ~]# systemctl restart httpd //起服务

[root@room9pc13 ~]# systemctl enable httpd //设置开机自启

2）确认前一步已经部署到Web网站目录的RHEL7光盘数据：

[root@room9pc13 ~]# du -sh /var/www/html/rh7dvd/ //检查部署结果

3.9G /var/www/html/rh7dvd/

3）将准备好的LibreOffice仓库目录部署到Web网页目录

[root@room9pc13 ~]# mv /opt/libreoffice/ /var/www/html/

[root@room9pc13 ~]# du -sh /var/www/html/llibreoffice/ //检查部署结果

234M /var/www/html/libreoffice

### 步骤二：在pc207上访问yum仓库

1）添加新的yum仓库设置

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

[rh7dvd]

name = RHEL 7.2 Server

baseurl = http://192.168.4.254/rh7dvd

gpgcheck = 0

[libreoffice]

name = LibreOffice 5

baseurl=http://192.168.4.254/libreoffice

gpgcheck = 0

2）测试新的yum仓库

[root@pc207 ~]# yum repolist

.. ..

repo id repo name status

libreoffice LibreOffice 5 53

rh7dvd RHEL 7.2 Server 4620

.. ..

## 6 案例6：vim效率操作

6.1 问题

本例要求掌握使用vim文本编辑器时能够提高操作效率的一些常用技巧和方法，完成下列任务：

将文件 /etc/passwd 复制为 /opt/nsd.txt，然后打开 /opt/nsd.txt 文件，练习命令模式下的切换/复制/删除/查找操作

将文件 /etc/man\_db.conf 复制到 /opt 目录下，然后打开 /opt/man\_db.conf 文件，将第50~100行内的“man”替换为“MAN”，在 vim 中设置显示行号查看效果

### 命令模式常用操作：

1G 或 gg ，跳转到文件的首行

G ，跳转到文件的末尾行

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^、d$ ，从光标处之前删除至行首/行尾

/word 向后查找字符串“word”，再按n/N跳至后/前一个结果

u ，撤销最近的一次操作

U ，撤销对当前行的所有修改

Ctrl + r 取消前一次撤销操作，与u相反

ZZ 保存修改并退出

Home 键 或 ^、数字 0 跳转到行首

End 键 或“$”键 跳转到行尾

PgUp 键、PgDn 键 向上翻页、向下翻页

### 末行模式常用操作：

:wq保存并退出 :w保存当前 :q!强制退出不保存

:s/old/new ，替换当前行第一个“old”

:s/old/new/g ，替换当前行所有的“old”

:n,m s/old/new/g ，替换第n-m行所有的“old”

:1,20 s/old/new/g 替换第1-20行所有的“old”

:% s/old/new/g ，替换文件内所有的“old”

:w /root/newfile ，另存为其它文件

:r /etc/filesystems ，读入其他文件内容

:set nu|nonu ，显示/不显示行号

:set ai|noai ，启用/关闭自动缩进

### 步骤一：vim命令模式下的切换/复制/删除/查找

1）建立练习文件:将文件 /etc/passwd 复制为 /opt/nsd.txt：

[root@svr7 ~]# cp /etc/passwd /opt/nsd.txt

2）使用vim打开练习文件，默认处于命令模式

[root@svr7 ~]# vim /opt/nsd.txt

3）在命令模式下完成下列操作

切换操作：G 最后一行，5G 第5行，gg 第一行。

复制操作：按2yy复制2行，7G移动到第7行，p 粘贴。

删除操作：25G 移动到第25行，200dd 从此行开始删除200行（不够就剩下全删）。

查找操作：gg 第一行，/adm 查找关键词adm，n 跳转到下一个结果。

4）保存并退出编辑器

ZZ 保存退出。

### 步骤二：vim末行模式下的替换/设置操作

1）建立练习文件:将文件 /etc/man\_db.conf 复制到 /opt/ 目录下：

[root@svr7 ~]# cp /etc/man\_db.conf /opt/

2）使用vim打开练习文件，输入:切换到末行模式

[root@svr7 ~]# vim /opt/man\_db.conf

.. ..

:

3）在末行模式下完成下列操作

输入 :set nu ，确认后显示行号。

输入 :50,100 s/man/MAN/g ，确认将第50~100行内的“man”替换为“MAN”。

4）保存并退出编辑器

输入 :wq ，确认后保存并退出编辑器。

vim补充:

2.vim缓存文件(交换文件)

E325: 注意

发现交换文件 "/opt/.1.txt.swp"

[root@svr7 ~]# rm -rf /opt/.1.txt.swp

[root@svr7 ~]# vim /opt/1.txt

## 7 案例7：编译安装软件包

7.1 问题

本例要求掌握常规源代码应用的安装过程，通过编译的方式安装inotify-tools 软件工具，完成下列任务：释放 inotify-tools-3.13.tar.gz 源码包

配置 ./configure

编译 make、安装 make install

测试inotifywait监控工具的用法及用途

7.2 方案

对于标准源码发布的C/C++软件包，编译安装一般包括以下过程：

解包：使用tar命令，将下载的源代码释放至指定目录

配置：执行源码目录内的 ./configure 脚本，指定安装目录/功能模块等选项

编译：在源码目录下执行 make 操作，根据配置清单Makefile生成可执行的二进制程序文件

安装：在源码目录下执行make install 操作，将编译好的程序及相关文件复制到安装目录

### 步骤一：确认已配置好编译环境

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

.. ..

[root@svr7 ~]# gcc --version

gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-4)

Copyright (C) 2015 Free Software Foundation, Inc.

This is free software; see the source for copying conditions. There is NO

warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 步骤二：编译安装inotify-tools软件包

1）解包inotify-tools-3.13.tar.gz文件

[root@svr7 ~]# ls inotify-tools-3.13.tar.gz

inotify-tools-3.13.tar.gz

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

2）配置 ./configure，安装目录默认（/usr/local/\*/）

[root@svr7 ~]# cd /usr/src/inotify-tools-3.13/ //进入源码目录

[root@svr7 inotify-tools-3.13]# ./configure //配置操作

checking for a BSD-compatible install... /usr/bin/install -c

checking whether build environment is sane... yes

checking for gawk... gawk

.. ..

configure: creating ./config.status

config.status: creating Makefile

.. ..

[root@svr7 inotify-tools-3.13]# ls Makefile //检查配置结果

Makefile

3）编译 make

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

.. ..

Making all in src

make[2]: Entering directory `/usr/src/inotify-tools-3.13/src'

make[3]: Entering directory `/usr/src/inotify-tools-3.13'

make[3]: Leaving directory `/usr/src/inotify-tools-3.13'

.. ..

4）安装 make install

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

.. ..

/usr/bin/install -c .libs/inotifywait /usr/local/bin/inotifywait

/bin/sh ../libtool --mode=install /usr/bin/install -c 'inotifywatch' '/usr/local/bin/inotifywatch'

.. ..

[root@svr7 inotify-tools-3.13]# find /usr/local/ -name "inotify\*"

/usr/local/bin/inotifywait //确认安装结果

/usr/local/bin/inotifywatch

/usr/local/include/inotifytools

/usr/local/include/inotifytools/inotifytools.h

## 8 inotifywait文档监控程序

软件包inotify-tools提供了一个主要程序inotifywait，可以用来监控指定目录或文档的变化，并及时给出通知。

1）开启对/opt目录的事件监控

[root@svr7 ~]# inotifywait -mrq /opt & //开启监控并放入后台运行

[1] 15568

2）修改/opt/目录内容，观察屏幕输出信息

[root@svr7 ~]# touch /opt/a.txt //新建文件a.txt

/opt/ CREATE a.txt

/opt/ OPEN a.txt

/opt/ ATTRIB a.txt

/opt/ CLOSE\_WRITE,CLOSE a.txt

[root@svr7 ~]# mv /opt/a.txt /opt/b.txt //将文件改名

/opt/ MOVED\_FROM a.txt

/opt/ MOVED\_TO b.txt

3）结束inotifywait监控

杀死当前用户的第一个后台任务：

[root@svr7 ~]# kill -9 %1

[1]+ Killed inotifywait -mrq /opt

## 9 案例8：使用systemctl工具

8.1 问题

本例要求掌握systemctl控制工具的基本操作，完成下列任务：

重启 httpd、crond、bluetooth 服务，查看状态

禁止 bluetooth 服务开机自启，并停用此服务

设置默认级别为 multi-user.target 并确认

8.2 方案

systemd是一个更高效的系统&服务管理器，其相关特性如下：

开机服务并行启动，各系统服务间的精确依赖

配置目录：/etc/systemd/system/

服务目录：/lib/systemd/system/

systemctl是systemd的管理工具，将相关资源组织为unit配置单元进行管理。

不同的unit决定了一组相关的启动任务，service和target是最常用的配置单元：

service：后台独立服务

target：一套配置单元的组合，类似于传统“运行级别”

### 1启动系统服务

启动系统服务httpd、crond、bluetooth

[root@svr7 ~]# systemctl start httpd crond bluetooth

### 2重启系统服务

重启系统服务httpd、crond、bluetooth

[root@svr7 ~]# systemctl restart httpd crond bluetooth

### 3查看服务的状态

[root@svr7 ~]# systemctl status httpd crond bluetooth

\* httpd.service - The Apache HTTP Server

Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)

Active: active (running) since Fri 2017-01-06 18:18:20 CST; 18s ago

.. ..

\* crond.service - Command Scheduler

Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)

Active: active (running) since Fri 2017-01-06 18:18:19 CST; 19s ago

.. ..

\* bluetooth.service - Bluetooth service

Loaded: loaded (/usr/lib/systemd/system/bluetooth.service; enabled; vendor preset: enabled)

Active: active (running) since Fri 2017-01-06 18:18:19 CST; 19s ago

### 4停用服务

bluetooth

[root@svr7 ~]# systemctl stop bluetooth

### 5禁止服务开机自启

bluetooth

[root@svr7 ~]# systemctl disable bluetooth

Removed symlink /etc/systemd/system/dbus-org.bluez.service.

Removed symlink /etc/systemd/system/bluetooth.target.wants/bluetooth.service.

[root@svr7 ~]# systemctl is-enabled Bluetooth //检查结果

disabled

### 运行级别

#### 1）查看默认运行级别

[root@svr7 ~]# systemctl get-default

graphical.target

#### 2）将默认运行级别设置为multi-user.target

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

Removed symlink /etc/systemd/system/default.target.

Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/multi-user.target.

#### 3）确认配置结果

[root@svr7 ~]# systemctl get-default

multi-user.target

根据此处的设置，重启此虚拟机后图形桌面将不再可用。

# 系统与服务管理进阶SERVICES02

## DNS解析的作用

• 为什么需要DNS系统

– www.baidu.com 与 119.75.217.56,哪个更好记?

– 互联网中的114查号台/导航员

• 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

所有的域名必须以点结尾 www.baidu.com.

根域名: .

一级域名: .cn .us .hk .tw .jp .kr ........

二级域名: .com.cn .edu.cn .org.cn .net.cn .........

三级域名: nb.com.cn haha.com.cn xixi.com.cn ......

完整合格的主机名: www.nb.com.cn ftp.nb.com.cn

完整合格的主机名: 主机头部+域名

• Full Qualified Domain Name(FQDN),完全合格主机名

– = 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/

[root@svr7 ~]# yum -y install bind bind-chroot

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)

– 系统服务:named

– DNS默认端口:TCP/UDP 53

– 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/ #主机名与IP地址对应关系

### 课堂案例1（第一个解析）

要求如下：

fsvr7.tedu.cn ---> 解析后返回192.168.4.7

www.tedu.cn --->解析后返回 192.168.4.1

ftp.tedu.cn ---> 解析后返回192.168.4.2

#### 1.服务端安装软件包

[root@svr7 ~]# yum -y install bind bind-chroot

#### 2.修改主配置文件

[root@svr7 ~]# cp /etc/named.conf /etc/named.bak #先备份原配置文件再修改

[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 { #指定负责解析的域名为tedu.cn

type master; #指定本机为权威服务器

file "tedu.cn.zone"; #指定地址库文件名称

};

#### 建立地址库文件

新建/var/named/tedu.cn.zone地址库文件，保证named用户有读权限

]# cd /var/named

]# cp -p named.localhost tedu.cn.zone

#拷贝一个模板，改名，-P继承被拷贝文件的属组root用户拷贝的文件，属主属组都为root

]# ls -l tedu.cn.zone

]# vim /var/named/tedu.cn.zone

#所有域名都必须以点结尾

#不以点结尾,自动补全本地址库文件负责的域名

tedu.cn. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.1

ftp A 192.168.4.2

#### 4.重起named服务

[root@svr7 /]# systemctl restart named

#### 5.客户端测试

1.指定本机DNS服务器地址，配置/etc/resolv.conf文件

]# echo nameserver 192.168.4.7 > /etc/resolv.conf

]# cat /etc/resolv.conf

nameserver 192.168.4.7

2.测试解析

[root@pc207 ~]# nslookup www.tedu.cn

[root@pc207 ~]# nslookup [ftp.tedu.cn](ftp://ftp.tedu.cn)

### 课堂案例2（新增一个解析）

虚拟机A:部署DNS服务,最终实现 www.qq.com ----> 1.2.3.4

#### 1.修改主配置文件（增加一个zone）

[root@svr7 ~]# vim /etc/named.conf

options {

directory "/var/named";

};

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "qq.com" IN { #添加一个zone

type master;

file "qq.com.zone";

};

#### 2.建立地址库文件

新建地址库文件/var/named/qq.com.zone #保证named用户有读权限

]# cd /var/named

]# cp -p tedu.cn.zone qq.com.zone #拷贝已有zone文件并保持权限进行复制

]# ls -l qq.com.zone

]# vim /var/named/qq.com.zone

qq.com. NS svr7

svr7 A 192.168.4.7

www A 1.2.3.4

#### 3.重起named服务

[root@svr7 /]# systemctl restart named

### DNS轮询与泛域名解析

2.1 问题

沿用案例1，本例要求掌握DNS轮询、泛域名解析的配置，实现的目标如下：

为站点 www.tedu.cn 提供DNS轮询解析，三台Web服务器节点的IP地址分别为：192.168.4.100、192.168.4.110、192.168.4.120

配置泛域名解析实现以下解析记录：任意名称.tedu.cn ---> 119.75.217.56

2.2 方案

DNS轮询：一个FQDN --->解析 多个地址IP地址1、IP地址2、.. ..

泛域名解析（站点名不确定）：多个FQDN ---> 解析一个IP地址

#### 步骤一：配置DNS轮询

1）修改DNS服务器上tedu.cn区域的地址库文件，在末尾添加轮询地址记录

[root@svr7 ~]# vim /var/named/tedu.cn.zone

.. ..

www A 192.168.4.100

www A 192.168.4.110

www A 192.168.4.120

2）重启系统服务named

[root@svr7 named]# systemctl restart named

3）在客户机pc207上测试轮询记录

针对目标www.tedu.cn执行多次查询，观察第1条结果的变化：

[root@pc207 ~]# host www.tedu.cn

www.tedu.cn has address 192.168.4.100 //第1个结果为192.168.4.100

www.tedu.cn has address 192.168.4.110

www.tedu.cn has address 192.168.4.120

[root@pc207 ~]# host www.tedu.cn

www.tedu.cn has address 192.168.4.120 //第1个结果为192.168.4.120

www.tedu.cn has address 192.168.4.110

www.tedu.cn has address 192.168.4.100

[root@pc207 ~]# host www.tedu.cn

www.tedu.cn has address 192.168.4.110 //第1个结果为192.168.4.110

www.tedu.cn has address 192.168.4.120

www.tedu.cn has address 192.168.4.100

#### 步骤二：配置多对一的泛域名解析

1）修改DNS服务器上指定区域的地址库文件，在末尾添加\*通配地址记录

[root@svr7 ~]# vim /var/named/tedu.cn.zone

.. ..

\* A 119.75.217.56

2）重启系统服务named

[root@svr7 named]# systemctl restart named

3）在客户机pc207上测试多对一的泛域名解析记录

当查询未知站点（地址库中没有明确记录）时，以 \* 对应的IP地址反馈：

[root@pc207 ~]# host station123.tedu.cn

station123.tedu.cn has address 119.75.217.56

[root@pc207 ~]# host movie.tedu.cn

movie.tedu.cn has address 119.75.217.56

[root@pc207 ~]# host tts8.tedu.cn

tts8.tedu.cn has address 119.75.217.56

### 配置DNS子域授权

3.1 问题

沿用案例1，本例要求为上下级两个DNS区域建立父子关联，实现客户机向父DNS也可以查询到子域内的FQDN，基本要求如下：

构建父DNS（tedu.cn）服务器

构建子DNS（bj.tedu.cn）服务器

在父DNS上配置子域授权

测试子域授权查询

3.2 方案

为一个DNS区域添加授权子域时，需要修改此区域的地址库，添加以下记录：

子域域名. IN NS 子DNS的FQDN.

子DNS的FQDN. IN A 子DNS的IP地址

3.3 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：构建父DNS（tedu.cn）服务器

1）将svr7配置为父DNS服务器，确认配置

主配置文件/etc/named.conf：

[root@svr7 ~]# viim /etc/named.conf

options {

directory "/var/named";

};

zone "tedu.cn" {

type master;

file "tedu.cn.zone";

};

.. ..

2）配置正向地址库文件：

[root@svr7 ~]# vim /var/named/tedu.cn.zone

$TTL 1D

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

@ NS svr7.tedu.cn.

svr7 A 192.168.4.7

pc207 A 192.168.4.207

www A 192.168.4.100

.. ..

3）确保服务已启用：

[root@svr7 ~]# systemctl restart named

4）测试 —— 向父DNS可成功查询到父区域中的站点

[root@pc207 ~]# host www.tedu.cn 192.168.4.7

Using domain server:

Name: 192.168.4.7

Address: 192.168.4.7#53

Aliases:

www.tedu.cn has address 192.168.4.100

.. ..

#### 步骤二：构建子DNS（bj.tedu.cn）服务器

1）将pc207配置为子DNS服务器，确认配置

安装软件包bind、bind-chroot：

[root@pc207 ~]# yum -y install bind bind-chroot

.. ..

2）建立主配置文件/etc/named.conf：

[root@pc207 ~]# mv /etc/named.conf /etc/named.conf.origin //备份默认配置

[root@pc207 ~]# vim /etc/named.conf //建立新配置

options {

directory "/var/named";

};

zone "bj.tedu.cn" { //定义子DNS的正向区域

type master;

file "bj.tedu.cn.zone";

};

3）建立地址库配置文件：

[root@pc207 ~]# cd /var/named/ //进地址库目录

[root@pc207 named]# cp -p named.localhost tedu.cn.zone //参考范本建地址库文件

[root@pc207 named]# vim bj.tedu.cn.zone //修订地址库记录

$TTL 1D //文件开头部分可保持不改

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

@ NS pc207.bj.tedu.cn. //本区域DNS服务器的FQDN

pc207 A 192.168.4.207 //为NS主机提供A记录

www A 1.2.3.4 //添加测试记录 www.bj.tedu.cn

4）启动系统服务named，并设置开机自启

[root@pc207 named]# systemctl restart named

[root@pc207 named]# systemctl enable named

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

5）测试 —— 向子DNS可成功查询到子区域中的站点

[root@pc207 ~]# host www.bj.tedu.cn 192.168.4.207

Using domain server:

Name: 192.168.4.207

Address: 192.168.4.207#53

Aliases:

www.bj.tedu.cn has address 1.2.3.4

#### 步骤三：在父DNS上配置子域授权

1）测试 —— 未配置子域授权时，向父DNS无法正确查询到子区域中的站点

若父DNS配置有 \* 泛域名，则反馈的结果为对应的IP地址119.75.217.56，而不是子DNS中记录的1.2.3.4：

[root@pc207 ~]# host www.bj.tedu.cn 192.168.4.7

Using domain server:

Name: 192.168.4.7

Address: 192.168.4.7#53

Aliases:

www.bj.tedu.cn has address 119.75.217.56

若父DNS未配置有 \* 泛域名，则找不到解析结果（not found）：

[root@pc207 ~]# host www.bj.tedu.cn 192.168.4.7

Using domain server:

Name: 192.168.4.7

Address: 192.168.4.7#53

Aliases:

Host www.bj.tedu.cn not found: 3(NXDOMAIN)

2）修改父DNS区域tedu.cn的地址库，添加授权子域信息

[root@svr7 ~]# vim /var/named/tedu.cn.zone

.. ..

bj.tedu.cn. NS pc207.bj.tedu.cn. //子区域及子DNS主机名

pc207.bj.tedu.cn. A 192.168.4.207 //子DNS的IP地址

[root@svr7 named]# systemctl restart named //重启服务

注意：pc207.bj.tedu.cn命名都无要求，见名知意就行

#### 步骤四：测试子域授权查询

父域：www.tedu.cn 在虚拟机server上

子域：www.bj.tedu.cn 在虚拟机pc207上

测试 —— 成功配置子域授权以后，向父DNS可以正确查询到子区域中的站点：

[root@pc207 ~]# host www.bj.tedu.cn 192.168.4.7

Using domain server:

Name: 192.168.4.7

Address: 192.168.4.7#53

Aliases:

www.bj.tedu.cn has address 1.2.3.4

客户端域名解析优先级：

、/etc/hosts > /etc/resolv.conf

## 搭建并测试缓存DNS

4.1 问题

本例要求熟悉缓存DNS的工作过程，准备一台可上网的RHEL7虚拟机，并完成下列任务：

安装 bind、bind-chroot 包

搭建并测试基于全局转发器的缓存DNS

注意：若所在机房不具备访问互联网DNS条件，此案例改由学员自行在家完成。

4.2 方案

#### 权威/官方DNS服务器的特点：

至少管理一个DNS区域,，需要IANA等官方机构授权

典型应用：根域DNS、一级域DNS、二级域DNS、三级域DNS、.. ..

#### 缓存DNS服务器的特点：

不需要管理任何DNS区域，但是能够替客户机查询，而且通过缓存、复用查询结果来加快响应速度

典型应用：ISP服务商、企业局域网

#### 缓存DNS服务器的解析记录来源：

方式1：全局转发：将请求转发给指定的公共DNS（其他缓存DNS），请求递归服务

方式2：根域迭代：依次向根、一级、二级……域的DNS服务器迭代

#### 步骤一：为虚拟机pc207提供上网条件

1）为虚拟机添加一块新的网卡，选择NAT或Bridge模式

若选择NAT模式（地址转换），则新加网卡的上网参数由虚拟化平台自动设置。

若选择Bridge模式（桥接），则新加网卡的上网参数需要参考真实网络的主机，必要时请网络管理员提供支持。

此处所列地址信息可帮助大家理解上网条件，但不作为练习的配置依据：

[root@pc207 ~]# ifconfig eth1 //检查新增网卡的IP地址

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.70.129 netmask 255.255.255.0 broadcast 192.168.70.255

.. ..

[root@pc207 ~]# route -n //确认已配好默认网关

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

0.0.0.0 192.168.70.2 0.0.0.0 UG 100 0 0 eth1

192.168.70.0 0.0.0.0 255.255.255.0 U 100 0 0 eth1

.. ..

[root@pc207 ~]# cat /etc/resolv.conf //确认第一DNS为外部可用DNS地址

nameserver 192.168.70.2

.. ..

2）确保从主机pc207可访问到外部DNS

访问默认DNS可用（本机正常连网需要）：

[root@pc207 ~]# host www.qq.com

www.qq.com has address 111.30.132.101

www.qq.com has IPv6 address 240e:e1:8100:28::2:16

访问指定DNS可用（全局转发的前提条件）：

[root@pc207 ~]# host www.qq.com 202.106.0.20 //国内公共DNS服务器之一

Using domain server:

Name: 202.106.0.20

Address: 202.106.0.20#53

Aliases:

www.qq.com has address 111.30.132.101

www.qq.com is an alias for qq.com.edgesuite.net.

qq.com.edgesuite.net is an alias for a1574.b.akamai.net.

www.qq.com is an alias for qq.com.edgesuite.net.

qq.com.edgesuite.net is an alias for a1574.b.akamai.net.

#### 步骤二：将pc207配置为缓存DNS（全局转发式）

1）安装bind、bind-chroot软件包

[root@pc207 ~]# yum -y install bind bind-chroot

.. ..

2）建立主配置文件/etc/named.conf

当收到来自客户机的DNS查询请求时，转发到外网的其他DNS服务器

[root@pc207 ~]# vim /etc/named.conf

options {

forwarders { 202.106.0.20; };

};

3）启动系统服务named，并设置开机自启

[root@pc207 ~]# systemctl restart named

[root@pc207 ~]# systemctl enable named

4）可向缓存DNS服务器pc207查询到公共域名（百度、网易等站点）

[root@pc207 ~]# host www.baidu.com 192.168.4.207 //查百度的站点IP

Using domain server:

Name: 192.168.4.207

Address: 192.168.4.207#53

Aliases:

www.baidu.com is an alias for www.a.shifen.com.

www.a.shifen.com has address 111.13.100.92

www.a.shifen.com has address 111.13.100.91

[root@pc207 ~]# host www.163.com 192.168.4.207 //查网易的站点IP

Using domain server:

Name: 192.168.4.207

Address: 192.168.4.207#53

Aliases:

www.163.com is an alias for www.163.com.lxdns.com.

www.163.com.lxdns.com is an alias for 163.xdwscache.ourglb0.com.

163.xdwscache.ourglb0.com has address 111.11.31.104

163.xdwscache.ourglb0.com has address 111.11.31.114

# 系统与服务管理进阶SERVICES03

## 配置并验证Split分离解析

1.1 问题

本例要求配置一台智能DNS服务器，针对同一个FQDN，当不同的客户机来查询时能够给出不同的答案。需要完成下列任务：

从主机192.168.4.207查询时，结果为：www.tedu.cn ---> 192.168.4.100

从其他客户端查询时，www.tedu.cn ---> 1.2.3.4

1.2 方案

在配置DNS服务器时，通过view视图设置来区分不同客户机、不同地址库：

view "视图1" {

match-clients { 客户机地址1; .. .. ; }; //匹配第1类客户机地址

zone "目标域名" IN { //同一个DNS区域

type master;

file "地址库1"; //第1份地址库

};

};

view "视图2" {

match-clients { 客户机地址2; .. .. ; }; //匹配第2类客户机地址

match-clients { any; }; //匹配任意地址

zone "目标域名" IN { //同一个DNS区域

type master;

file "地址库2"; //第2份地址库

};

};

.. ..

view "视图n" {

match-clients { any; }; //匹配任意地址

zone "目标域名" IN { //同一个DNS区域

type master;

file "地址库n"; //第n份地址库

};

};

### 步骤一：配置Split分离解析

1）为tedu.cn区域建立两份解析记录文件

第一份解析记录文件提供给客户机192.168.4.207、网段192.168.7.0/24，对应目标域名www.tedu.cn的A记录地址为192.168.4.100。相关操作及配置如下：

[root@svr7 ~]# cd /var/named/

[root@svr7 named]# cp -p tedu.cn.zone tedu.cn.zone.lan

[root@svr7 named]# vim tedu.cn.zone.lan

$TTL 1D

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

@ NS svr7.tedu.cn.

svr7 A 192.168.4.7

pc207 A 192.168.4.207

www A 192.168.4.100

第二份解析记录文件提供给其他客户机，对应目标域名www.tedu.cn的A记录地址为1.2.3.4。相关操作及配置如下：

[root@svr7 named]# cp -p tedu.cn.zone tedu.cn.zone.other

[root@svr7 named]# vim tedu.cn.zone.other

$TTL 1D

@ IN SOA @ rname.invalid. (

0 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

@ NS svr7.tedu.cn.

svr7 A 192.168.4.7

pc207 A 192.168.4.207

www A 1.2.3.4

2）修改named.conf配置文件，定义两个view，分别调用不同解析记录文件

[root@svr7 ~]# vim /etc/named.conf

options {

directory "/var/named";

};

acl "mylan" { //名为mylan的列表。名字见名知意即可

192.168.4.207; 192.168.7.0/24;

};

.. ..

view "mylan" {

match-clients { mylan; }; //检查客户机地址是否匹配此列表

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone.lan";

};

};

view "other" {

match-clients { any; }; //匹配任意客户机地址

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone.other";

};

};

3）重启named服务

[root@svr7 ~]# systemctl restart named

### 步骤二：测试分离解析效果

1）从mylan地址列表中的客户机查询

在客户机192.168.4.207（或网段192.168.7.0/24内的任意客户机）上查询www.tedu.cn，结果是 192.168.4.100：

[root@pc207 ~]# host www.tedu.cn 192.168.4.7

Using domain server:

Name: 192.168.4.7

Address: 192.168.4.7#53

Aliases:

www.tedu.cn has address 192.168.4.100

2）从其他客户机查询

在DNS服务器本机或CentOS真机上查询www.tedu.cn时，结果为 1.2.3.4：

[root@svr7 ~]# host www.tedu.cn 192.168.4.7

Using domain server:

Name: 192.168.4.7

Address: 192.168.4.7#53

Aliases:

www.tedu.cn has address 1.2.3.4

### 课堂实际配置配置Split分离解析：

修改named.conf配置文件，定义两个view，分别调用不同解析记录文件

[root@svr7 ~]# vim /etc/named.conf

options {

directory "/var/named";

};

view "mylan" {

match-clients { 192.168.4.207; 192.168.7.0/24; }; //与以上步骤不同之处

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone.lan";

};

};

view "other" {

match-clients { any; }; //匹配任意客户机地址

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone.other";

};

};

与步骤一不同之处：直接将匹配条件写入match-clients中，没有acl列表。

acl "mylan" { //名为mylan的列表。名字见名知意即可

192.168.4.207; 192.168.7.0/24;

};

.. ..

view "mylan" {

match-clients { mylan; }; //检查客户机地址是否匹配此列表

……

### acl 地址列表--类似地址变量

变量：以不变的名称存储变化的值

acl abc { 192.168.4.207; 192.168.7.0/24; }

view "nsd" {

match-clients { abc; }; #这两个地址使用这个地址库

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

};

#使用了acl 之后的效果如下

view "nsd" {

match-clients { 192.168.4.207; 192.168.7.0/24; }; #这两个地址使用这个地址库

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

};

===============================================

### 多区域的分离解析

每个view 中有多个zone，zone的个数与负责的域名保持一致，所使用的地址库文件不同

匹配由上往下，匹配即停止，匹配为any的要放最后

## RAID磁盘阵列

配置界面是英文界面，各个厂家出厂的服务器RAID配置可能不一样，借助售后配置

廉价冗余磁盘阵列：通过硬件/软件技术，将多个较小/低速的磁盘整合成一个大磁盘

阵列的价值：提升I/O效率、硬件级别的数据冗余

RAID级别：不同级别功能和特性各不相同

硬件

RAID卡，硬件产品，类似于网卡，用于配置RAID，有带密码的和不带密码的，读取所有磁盘信息，

磁盘1磁盘2磁盘3---》RAID---》整体磁盘------》linux---->分区使用，划分逻辑卷

### 1、RAID0模式

优点：在RAID 0状态下，存储数据被分割成两部分，分别存储在两块硬盘上，此时移动硬盘的理论存储速度是单块硬盘的2倍，实际容量等于两块硬盘中较小一块硬盘的容量的2倍。

缺点：任何一块硬盘发生故障，整个RAID上的数据将不可恢复。

备注：存储高清电影比较适合。

### 2、RAID1模式

优点：此模式下，两块硬盘互为镜像。当一个硬盘受损时，换上一块全新硬盘(大于或等于原硬盘容量)替代原硬盘即可自动恢复资料和继续使用，移动硬盘的实际容量等于较小一块硬盘的容量，存储速度与单块硬盘相同。RAID 1的优势在于任何一块硬盘出现故障是，所存储的数据都不会丢失。

缺点：该模式可使用的硬盘实际容量比较小，仅仅为两颗硬盘中最小硬盘的容量。

备注：非常重要的资料，如数据库，个人资料，是万无一失的存储方案。

### 3、RAID 0+1模式

RAID 0+1是磁盘分段及镜像的结合，采用2组RAID0的磁盘阵列互为镜像，它们之间又成为一个RAID1的阵列。硬盘使用率只有50%，但是提供最佳的速度及可靠度。

### 4、RAID 3模式

RAID3是把数据分成多个“块”，按照一定的容错算法，存放在N+1个硬盘上，实际数据占用的有效空间为N个硬盘的空间总和，而第N+1个硬盘存储的数据是校验容错信息，当这N+1个硬盘中的其中一个硬盘出现故障时，从其它N个硬盘中的数据也可以恢复原始数据。

### 5、RAID 5模式

RAID5不对存储的数据进行备份，而是把数据和相对应的奇偶校验信息存储到组成RAID5的各个磁盘上，并且奇偶校验信息和相对应的数据分别存储于不同的磁盘上。当RAID5的一个磁盘数据发生损坏后，利用剩下的数据和相应的奇偶校验信息去恢复被损坏的数据。

### 6、RAID 10模式

RAID10最少需要4块硬盘才能完成。把2块硬盘组成一个RAID1，然后两组RAID1组成一个RAID0。虽然RAID10方案造成了50%的磁盘浪费，但是它提供了200%的速度和单磁盘损坏的数据安全性。

### 另外三种硬件快速硬件设置模式简介

       1、Clone模式

    克隆模式，磁盘全部数据一样，以最小硬盘的为准。

    2、Large模式

    硬盘容量简单相加，将几个硬盘变成一个硬盘，容量为几个硬盘容量之和，此模式下可以获得最大的硬盘空间。

    3、Normal模式

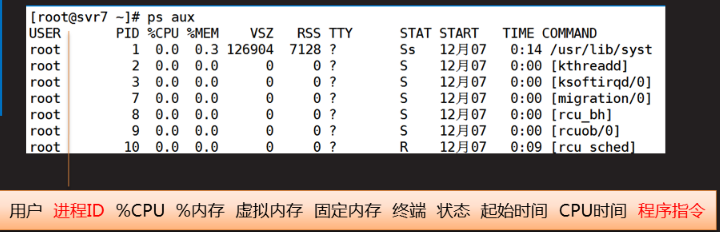
硬盘分别处于正常、独立的状态，可以分别独立的写入或读取资料，能使用的实际容量分别为4个硬盘的容量。如果其中一个硬盘受损，其他几个硬盘不会受影响。

## 进程管理主要命令工具：

### ps 查看进程快照

ps aux:查看进程静态快照

a显示当前终端所有进程 u以用户格式输出 x当前用户在所有终端下的进程



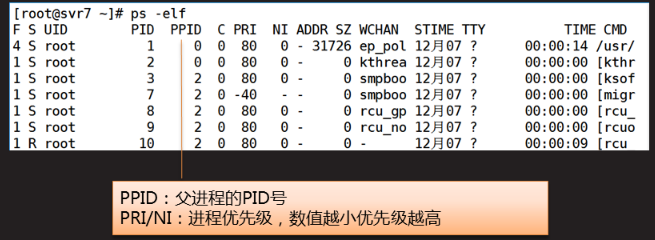
案例:

[root@svr7 ~]# ps aux | grep sleep

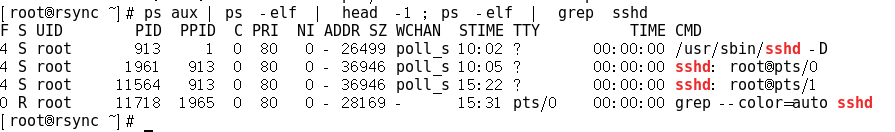
root 32929 0.0 0.0 4312 360 pts/1 S+ 17:25 0:00 sleep 600

ps –elf：查看进程静态快照

-e 显示系统内所有进程 -l以长格式输出信息 -f包括最完整的进程信息



[root@rsync ~]# ps -elf | head -1 ; ps -elf | grep sshd



### top 进程动态排名

top [-d 刷新秒数] [-U 用户名] ： 交互式工具,查看进程动态排名,

top打开之后按如下按键可显示不同内容:

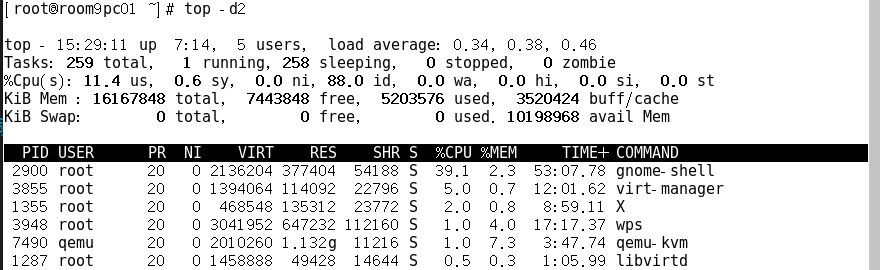
?: 查看帮助(列出可用的按键指令)

P:根据CPU降序排列 M:根据MEM降序排列

T:根据进程消耗的TIME降序排列

k:杀死指定的进程

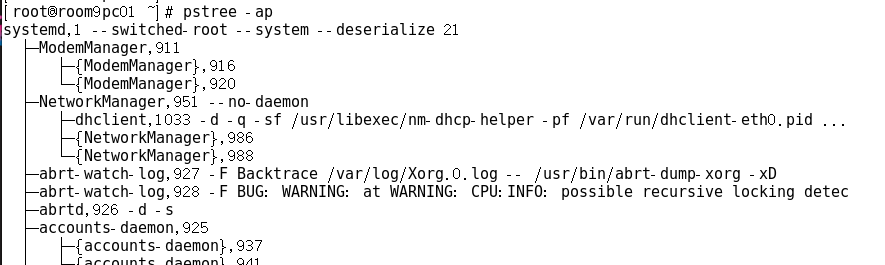
q:退出top进程

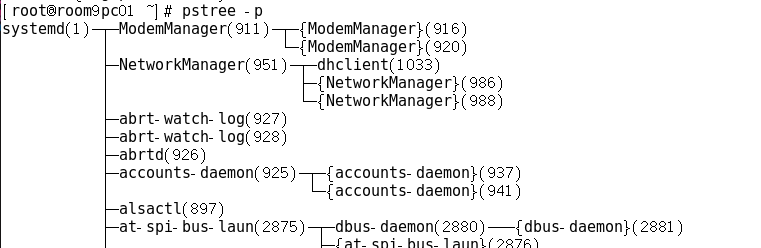


### pstree 查看进程树

pstree [选项] [PID或者用户名]：查看进程与进程之间的树型关系结构

-a :显示完整的命令行 -p:列出对应PID编号





### pgrep 检索进程

pgrep [选项]... 查询条件 ：根据指定的名称或条件检索进程

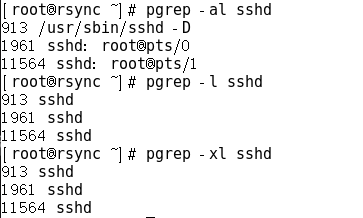
-l:输出进程名,而不仅仅是PID

-U:检索指定用户的进程

-t:检索指定终端的进程

-x:精确匹配完整的进程名

-a:



### 步骤一：找出进程 gdm 的 PID 编号值

使用pgrep命令查询指定名称的进程，选项-l显示PID号、-x精确匹配进程名：

[root@svr7 ~]# pgrep -lx gdm

1584 gdm

### 步骤二：列出由进程 gdm 开始的子进程树结构信息

使用pstree命令，可以提供用户名或PID值作为参数。通过前一步已知进程gdm的PID为1584，因此以下操作可列出进程gdm的进程树结构：

[root@svr7 ~]# pstree -p 1584

gdm(1584)-+-Xorg(1703)

|-gdm-session-wor(2670)-+-gnome-session(2779)-+-gnom+

| | |-gnom+

| | |-{gno+

| | |-{gno+

| | `-{gno+

| |-{gdm-session-wor}(2678)

| `-{gdm-session-wor}(2682)

|-{gdm}(1668)

|-{gdm}(1671)

`-{gdm}(1702)

### 步骤三：找出进程 sshd 的父进程的 PID 编号/进程名称

要查看进程的父进程PID，可以使用ps –elf命令，简单grep过滤即可。找到进程sshd所在行对应到的PPID值即为其父进程的PID编号。为了方便直观查看，建议先列出ps表头行，以分号隔开再执行过滤操作。

[root@svr7 ~]# ps -elf | head -1 ; ps -elf | grep sshd

F S UID PID PPID C PRI NI ADDR SZ WCHAN STIME TTY TIME CMD

4 S root 1362 1 0 80 0 - 20636 poll\_s Jan05 ? 00:00:00 /usr/sbin/sshd –D

.. .. //可获知进程sshd的父进程PID为1

然后再根据pstree –p的结果过滤，可获知PID为1的进程名称为systemd：

[root@svr7 ~]# pstree -p | grep '(1)'

systemd(1)-+-ModemManager(995)-+-{ModemManager}(1018)

### 步骤四：查看当前系统的CPU负载/进程总量信息

使用top命令，直接看开头部分即可；或者 top -n 次数：

[root@svr7 ~]# top

top - 15:45:25 up 23:55, 2 users, load average: 0.02, 0.03, 0.05

Tasks: 485 total, 2 running, 483 sleeping, 0 stopped, 0 zombie

%Cpu(s): 1.7 us, 1.0 sy, 0.0 ni, 97.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st

KiB Mem : 1001332 total, 76120 free, 419028 used, 506184 buff/cache

KiB Swap: 2097148 total, 2096012 free, 1136 used. 372288 avail Mem

.. ..

观察Tasks: 485 total部分，表示进程总量信息。

观察load average: 0.02, 0.03, 0.05 部分，表示CPU处理器在最近1分钟、5分钟、15分钟内的平均处理请求数（对于多核CPU，此数量应除以核心数）。

对于多核CPU主机，如果要分别显示每颗CPU核心的占用情况，可以在top界面按数字键1进行切换：

[root@svr7 ~]# top

top - 15:47:45 up 23:57, 2 users, load average: 0.02, 0.03, 0.05

Tasks: 485 total, 2 running, 269 sleeping, 0 stopped, 1 zombie

Cpu0 : 0.6%us, 7.8%sy, 0.0%ni, 91.6%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Cpu1 : 0.7%us, 3.7%sy, 0.0%ni, 95.6%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Cpu2 : 0.7%us, 1.7%sy, 0.0%ni, 97.6%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Cpu3 : 0.3%us, 1.0%sy, 0.0%ni, 98.3%id, 0.3%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 16230564k total, 15716576k used, 513988k free, 326124k buffers

Swap: 8388604k total, 220656k used, 8167948k free, 11275304k cached

.. ..

# 进程调度及终止

本例要求掌握调度及终止进程的操作，使用必要的工具完成下列任务：

运行“sleep 600”命令，再另开一个终端，查出sleep程序的PID并杀死

运行多个vim程序并都放入后台，然后杀死所有vim进程

su切换为zhsan用户，再另开一个终端，强制踢出zhsan用户

进程调度及终止的主要命令工具：

* 命令行 &：将命令行在后台运行
* Ctrl + z 组合键：挂起当前进程（暂停并转入后台）
* jobs：列出当前用户当前终端的后台任务
* bg 编号：启动指定编号的后台任务
* fg 编号：将指定编号的后台任务调入前台运行
* kill [-9] PID...：杀死指定PID值的进程
* kill [-9] %n：杀死第n个后台任务
* killall [-9] 进程名...：杀死指定名称的所有进程
* pkill：根据指定的名称或条件杀死进程

实现此案例需要按照如下步骤进行。

### 步骤一：根据PID杀死进程

1）开启sleep测试进程

[root@svr7 ~]# sleep 600

//.. .. 进入600秒等待状态

2）找出进程sleep的PID

另开一个终端，ps aux并过滤进程信息（第2列为PID值）：

[root@svr7 ~]# ps aux | grep sleep

root 32929 0.0 0.0 4312 360 pts/1 S+ 17:25 0:00 sleep 600

3）杀死指定PID的进程

[root@svr7 ~]# kill -9 32929

返回原终端会发现sleep进程已经被杀死：

[root@svr7 ~]# sleep 600

Killed

### 步骤二：根据进程名杀死多个进程

1）在后台开启多个vim进程

[root@svr7 ~]# vim a.txt &

[1] 33152

[root@svr7 ~]# vim b.txt &

[2] 33154

[1]+ 已停止 vim a.txt

[root@svr7 ~]# vim c.txt &

[3] 33155

[2]+ 已停止 vim b.txt

2）确认vim进程信息

[root@svr7 ~]# jobs -l

[1] 33152 停止 (tty 输出) vim a.txt

[2]- 33154 停止 (tty 输出) vim b.txt

[3]+ 33155 停止 (tty 输出) vim c.txt

3）强制杀死所有名为vim的进程

[root@svr7 ~]# killall -9 vim

[1] 已杀死 vim a.txt

[2]- 已杀死 vim b.txt

[3]+ 已杀死 vim c.txt

4）确认杀进程结果

[root@svr7 ~]# jobs -l

[root@svr7 ~]#

### 步骤三：杀死属于指定用户的所有进程

1）登入测试用户zhsan

[root@svr7 ~]# useradd zhsan

[root@svr7 ~]# su - zhsan

[zhsan@svr7 ~]$

2）另开一个终端，以root用户登入，查找属于用户zhsan的进程

[root@svr7 ~]# pgrep -u zhsan

33219

[root@svr7 ~]# pstree -up 33219 //检查进程树

bash(33219,zhsan)

3）强制杀死属于用户zhsan的进程

[root@svr7 ~]# pkill -9 -u zhsan

[root@svr7 ~]#

4）返回原来用户zhsan登录的终端，确认已经被终止

[zhsan@svr7 ~]$ 已杀死

[root@svr7 ~]#

# 系统日志分析

## 常见的系统日志及各自用途：

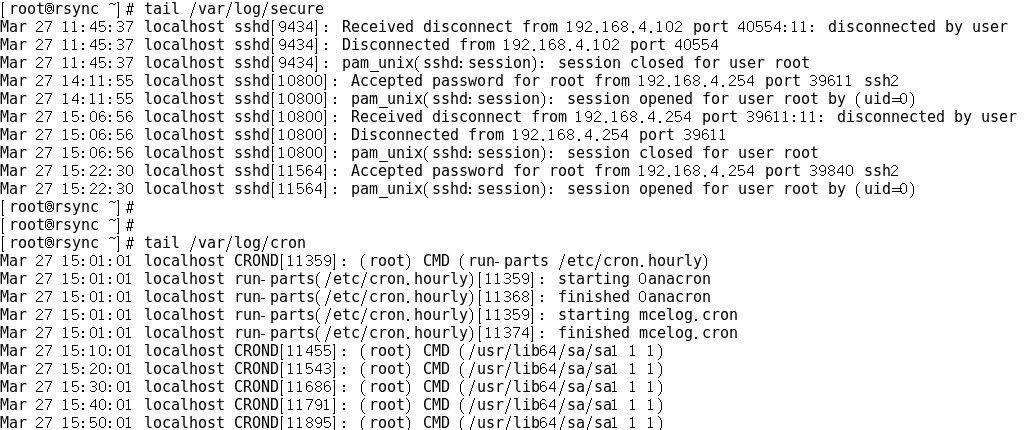
* /var/log/messages，记录内核消息、各种服务的公共消息
* /var/log/dmesg，记录系统启动过程的各种消息
* /var/log/cron，记录与cron计划任务相关的消息
* /var/log/maillog，记录邮件收发相关的消息
* /var/log/secure，记录与访问限制相关的安全消息,如远程登录等

由系统服务rsyslog统一记录管理:

日志消息采用文本格式

主要记录事件发生的时间,主机,进程,内容





## 用户日志:

/var/log/lastlog 记录最近的用户登录事件

/var/log/wtmp 记录成功的用户登录/注销事件

/var/log/btmp 记录失败的用户登录事件

/var/run/utmp 记录当前登录的每个用户的相关信息

由登录程序负责记录管理:

日志采用二进制格式

记录登录用户的时间来源执行的命令等

**用户登录分析:**

### 查看已登录的用户信息: users who w

[root@rsync ~]# users

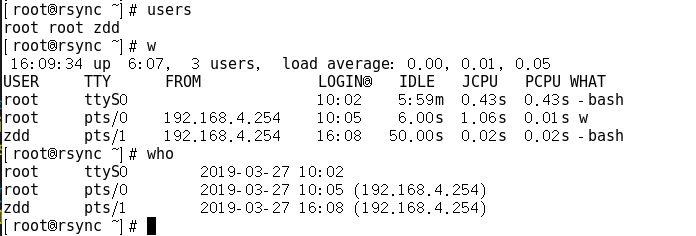
root root root #有3个终端,用root登录

[root@rsync ~]# users

root root #有2个终端,用root登录

[root@rsync ~]# users

root root zdd #有3个终端登录, 其中一个是zdd用户



### 查看最近登录成功(last)/失败(lastb)的用户信息

[root@rsync ~]# last -4 #登录成功的最近4条记录

zdd pts/1 192.168.4.254 Wed Mar 27 16:08 still logged in

root pts/1 192.168.4.254 Wed Mar 27 16:08 - 16:08 (00:00)

root pts/1 192.168.4.254 Wed Mar 27 15:22 - 16:04 (00:42)

root pts/1 192.168.4.254 Wed Mar 27 14:11 - 15:06 (00:55)

wtmp begins Tue Jan 30 12:24:54 2018

[root@rsync ~]# lastb -4 #登录失败的最近4条记录

root ssh:notty 192.168.4.102 Wed Mar 27 11:38 - 11:38 (00:00)

(unknown ttyS0 Wed Mar 27 10:02 - 10:02 (00:00)

root ttyS0 Wed Mar 27 10:02 - 10:02 (00:00)

(unknown ttyS0 Tue Jan 30 12:35 - 12:35 (00:00)

btmp begins Tue Jan 30 12:35:46 2018

## 日志消息的优先级（高-->低）：

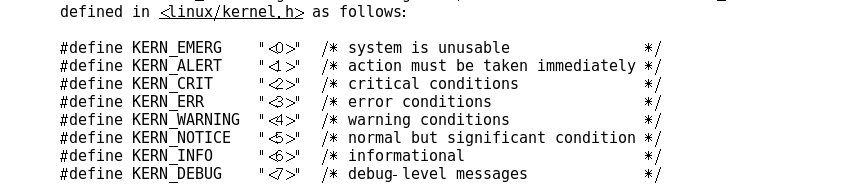
linux内核定义的事件紧急程度:

分为0~7共8种优先级

数值越小,表示对应事件越紧急/重要

* EMERG（紧急） ：级别0，系统不可用的情况
* ALERT（警报） ：级别1，必须马上采取措施的情况
* CRIT（严重） ：级别2，严重情形
* ERR（错误） ：级别3，出现错误
* WARNING（警告） ：级别4，值得警告的情形
* NOTICE（注意） ：级别5，普通但值得引起注意的事件
* INFO（信息） ：级别6，一般信息
* DEBUG（调试） ：级别7，程序/服务调试消息

[root@rsync ~]# man 2 syslog



## RHEL7提供的journalctl日志工具的常见用法：

提取由systemd-journal服务搜集的日志

主要包括内核/系统日志/服务日志

**常见用法:**

* journalctl | grep 关键词
* journalctl -u 服务名 -p 优先级
* journalctl -n 消息条数
* journalctl --since="yyyy-mm-dd HH:MM:SS" --until="yyyy-mm-dd HH:MM:SS"

### 步骤一：分析系统日志及用户日志

#### 1）列出所有包含关键词8909的系统日志消息

简单模拟一个故障（SELinux阻止Web开放8909端口）：

[root@svr7 ~]# vim /etc/httpd/conf.d/8909.conf //添加开8909端口配置

Listen 8909

[root@svr7 ~]# setenforce 1 //开启强制模式

[root@svr7 ~]# systemctl restart httpd //起服务失败

Job for httpd.service failed because the control process exited with error code. See "systemctl status httpd.service" and "journalctl -xe" for details.

从日志文件/var/log/messages中检索信息：

[root@svr7 ~]# grep 8909 /var/log/messages

Jan 6 17:53:48 svr7 setroubleshoot: SELinux is preventing /usr/sbin/httpd from name\_bind access on the tcp\_socket port 8909. For complete SELinux messages. run sealert -l 6d37b8f0-ab8a-4082-9295-c784f4f57190

Jan 6 17:53:48 svr7 python: SELinux is preventing /usr/sbin/httpd from name\_bind access on the tcp\_socket port 8909.#012#012\*\*\*\*\* Plugin bind\_ports (92.2 confidence) suggests \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#012#012If you want to allow /usr/sbin/httpd to bind to network port 8909#012Then you need to modify the port type.#012Do#012# semanage port -a -t PORT\_TYPE -p tcp 8909#012 where PORT\_TYPE is one of the following: http\_cache\_port\_t, http\_port\_t, jboss\_management\_port\_t, jboss\_messaging\_port\_t, ntop\_port\_t, puppet\_port\_t.#012#012\*\*\*\*\* Plugin catchall\_boolean (7.83 confidence) suggests \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#012#012If you want to allow nis to enabled#012Then you must tell SELinux about this by enabling the 'nis\_enabled' boolean.#012#012Do#012setsebool -P nis\_enabled 1#012#012\*\*\*\*\* Plugin catchall (1.41 confidence) suggests \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#012#012If you believe that httpd should be allowed name\_bind access on the port 8909 tcp\_socket by default.#012Then you should report this as a bug.#012You can generate a local policy module to allow this access.#012Do#012allow this access for now by executing:#012# grep httpd /var/log/audit/audit.log | audit2allow -M mypol#012# semodule -i mypol.pp#012

.. ..

使用完毕记得删除测试配置文件：

[root@svr7 ~]# rm -rf /etc/httpd/conf.d/8909.conf

[root@svr7 ~]# systemctl restart httpd

#### 2）查看启动时识别的鼠标设备信息

[root@svr7 ~]# dmesg | grep -i mouse

[ 1.020385] mousedev: PS/2 mouse device common for all mice

[ 1.249422] input: ImPS/2 Generic Wheel Mouse as /devices/platform/i8042/serio1/input/input2

[ 2.279665] usb 2-1: Product: VMware Virtual USB Mouse

[ 2.603999] input: VMware VMware Virtual USB Mouse as /devices/pci0000:00/0000:00:11.0/0000:02:00.0/usb2/2-1/2-1:1.0/input/input3

[ 2.604222] hid-generic 0003:0E0F:0003.0001: input,hidraw0: USB HID v1.10 Mouse [VMware VMware Virtual USB Mouse] on usb-0000:02:00.0-1/input0

#### 3）列出最近2条成功/不成功的用户登录消息

查看成功登录的事件消息：

[root@svr7 ~]# last -2

zhsan pts/2 192.168.4.207 Fri Jan 6 18:00 - 18:00 (00:00)

root pts/2 192.168.4.110 Fri Jan 6 17:26 - 17:59 (00:33)

wtmp begins Thu Aug 4 00:10:16 2016

查看失败登录的事件消息：

[root@svr7 ~]# lastb -2

anonymou ssh:notty 192.168.4.207 Fri Jan 6 18:00 - 18:00 (00:00)

anonymou ssh:notty 192.168.4.207 Fri Jan 6 18:00 - 18:00 (00:00)

btmp begins Fri Jan 6 18:00:34 2017

### 步骤二：使用journalctl日志提取工具

#### 1）列出最近10条重要程度在 ERR 及以上的日志消息

[root@svr7 ~]# journalctl -p err -n 10

-- Logs begin at Thu 2017-01-05 15:50:08 CST, end at Fri 2017-01-06 18:01:01 CST. --

Jan 06 14:56:57 svr7 setroubleshoot[23702]: SELinux is preventing /usr/sbin/vsftpd from getattr access on the file /rhel7/repodata/repomd.xml. For complete SELinux mes

Jan 06 14:56:57 svr7 setroubleshoot[23702]: SELinux is preventing /usr/sbin/vsftpd from read access on the file repomd.xml. For complete SELinux messages. run sealert

Jan 06 14:56:57 svr7 setroubleshoot[23702]: SELinux is preventing /usr/sbin/vsftpd from read access on the file repomd.xml. For complete SELinux messages. run sealert

Jan 06 14:56:57 svr7 setroubleshoot[23702]: SELinux is preventing /usr/sbin/vsftpd from lock access on the file /rhel7/repodata/repomd.xml. For complete SELinux messag

Jan 06 17:53:48 svr7 setroubleshoot[33743]: Plugin Exception restorecon\_source

Jan 06 17:53:48 svr7 setroubleshoot[33743]: SELinux is preventing /usr/sbin/httpd from name\_bind access on the tcp\_socket port 8909. For complete SELinux messages. run

Jan 06 17:53:53 svr7 setroubleshoot[33743]: SELinux is preventing /usr/sbin/httpd from name\_connect access on the tcp\_socket port 8909. For complete SELinux messages.

Jan 06 17:53:54 svr7 systemd[1]: Failed to start The Apache HTTP Server.

.. ..

lines 1-11/11 (END)

#### 2）列出所有与服务httpd相关的消息

[root@svr7 ~]# journalctl -u httpd

-- Logs begin at Thu 2017-01-05 15:50:08 CST, end at Fri 2017-01-06 18:01:01 CST. --

Jan 06 14:57:16 svr7 systemd[1]: Starting The Apache HTTP Server...

Jan 06 14:57:16 svr7 httpd[23812]: AH00557: httpd: apr\_sockaddr\_info\_get() failed for svr7

Jan 06 14:57:16 svr7 httpd[23812]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directi

Jan 06 14:57:16 svr7 systemd[1]: Started The Apache HTTP Server.

Jan 06 17:53:44 svr7 systemd[1]: Stopping The Apache HTTP Server...

Jan 06 17:53:46 svr7 systemd[1]: Starting The Apache HTTP Server...

Jan 06 17:53:46 svr7 httpd[33741]: AH00557: httpd: apr\_sockaddr\_info\_get() failed for svr7

.. ..

#### 3）列出前4个小时内新记录的日志

根据当前日期时间往前推4个小时，确定--since起始和--until结束时刻:

[root@svr7 ~]# journalctl --since "2017-01-06 14:11" --until "2017-01-06 18:11"

-- Logs begin at Thu 2017-01-05 15:50:08 CST, end at Fri 2017-01-06 18:10:01 CST. --

Jan 06 14:20:01 svr7 systemd[1]: Started Session 160 of user root.

Jan 06 14:20:01 svr7 CROND[22869]: (root) CMD (/usr/lib64/sa/sa1 1 1)

Jan 06 14:20:01 svr7 systemd[1]: Starting Session 160 of user root.

Jan 06 14:30:01 svr7 systemd[1]: Started Session 161 of user root.

Jan 06 14:30:01 svr7 CROND[23028]: (root) CMD (/usr/lib64/sa/sa1 1 1)

Jan 06 14:31:39 svr7 systemd[1]: Starting Session 162 of user root.

Jan 06 14:32:17 svr7 sshd[23046]: pam\_unix(sshd:session): session closed for user root

Jan 06 14:31:39 svr7 systemd[1]: Started Session 162 of user root.

Jan 06 14:31:39 svr7 sshd[23046]: pam\_unix(sshd:session): session opened for user root by (uid=0)

Jan 06 14:31:39 svr7 systemd-logind[985]: New session 162 of user root.

# 系统与服务管理进阶SERVICES04

一 部署DHCP服务器

动态主机配置协议

DHCO地址分配的四次会话，广播形式，先到先得

DISCOVERY --》OFFER --》REQUEST --》ACK

发现：客户机广播寻找DHCP服务器 --》录用：DHCP回应有IP--》请求：客户机同意使用--》确定

一个网络中只能有一台DHCP

网络装机时，将区域网络隔离，否则影响所有的以DHCP方式获取IP地址的主机

服务端基本概念

租期

===============================================

yum -y install dhcp

[root@svr7 ~]# vim /etc/dhcp/dhcpd.conf

1 #

2 # DHCP Server Configuration file.

3 # see /usr/share/doc/dhcp\*/dhcpd.conf.example --模板在此文件中

4 # see dhcpd.conf(5) man page

5 #

:set nu -------末行模式中

:r /usr/share/doc/dhcp\*/dhcpd.conf.example -------末行模式中

在末行模式中输入[:r 文件地址] 将文件内容写入到当前文本文件中，注意光标的位置

删除多余行 最终如下

1 #

2 # DHCP Server Configuration file.

3 # see /usr/share/doc/dhcp\*/dhcpd.conf.example

4 # see dhcpd.conf(5) man page

5 subnet 10.5.5.0 netmask 255.255.255.224 {

6 range 10.5.5.26 10.5.5.30;

7 option domain-name-servers ns1.internal.example.org;

8 option domain-name "internal.example.org";

9 option routers 10.5.5.1;

10 option broadcast-address 10.5.5.31;

11 default-lease-time 600;

12 max-lease-time 7200;

13 }

~

修改地址到如下

1 #

2 # DHCP Server Configuration file.u

3 # see /usr/share/doc/dhcp\*/dhcpd.conf.example

4 # see dhcpd.conf(5) man page

5 subnet 192.168.4.0 netmask 255.255.255.0 { -------指定分配的网段

6 range 192.168.4.20 192.168.4.100; --------指定分配的IP地址范围

7 option domain-name-servers 192.168.4.7; --------给客户端配置的DNS为：192.168.4.7

8 option routers 192.168.4.254; --------给客户端配置的网关为：192.168.4.7

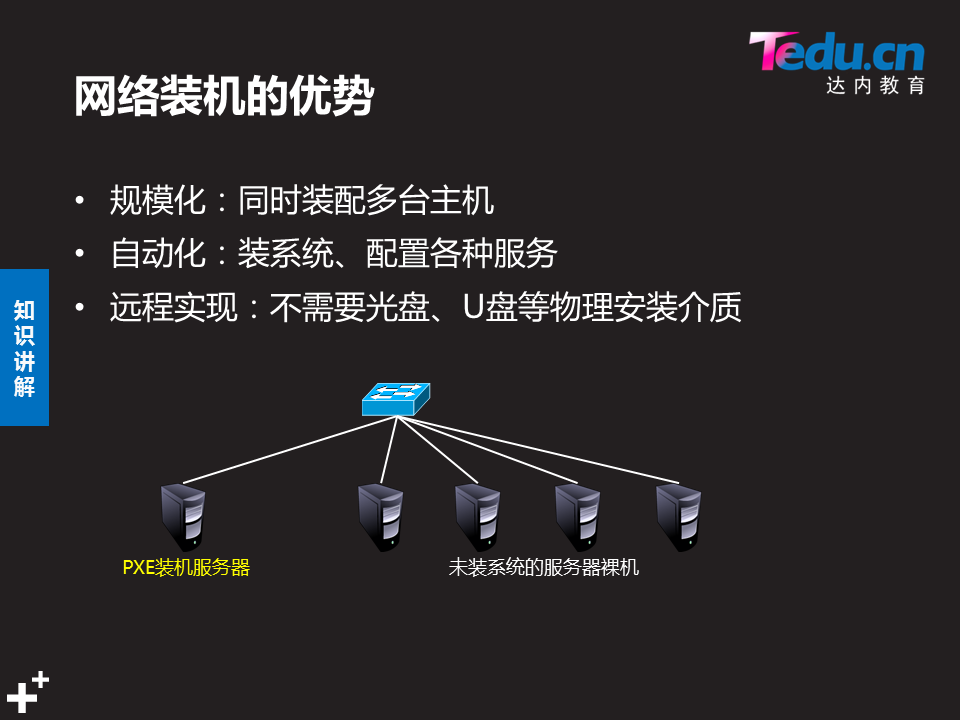
9 default-lease-time 600;

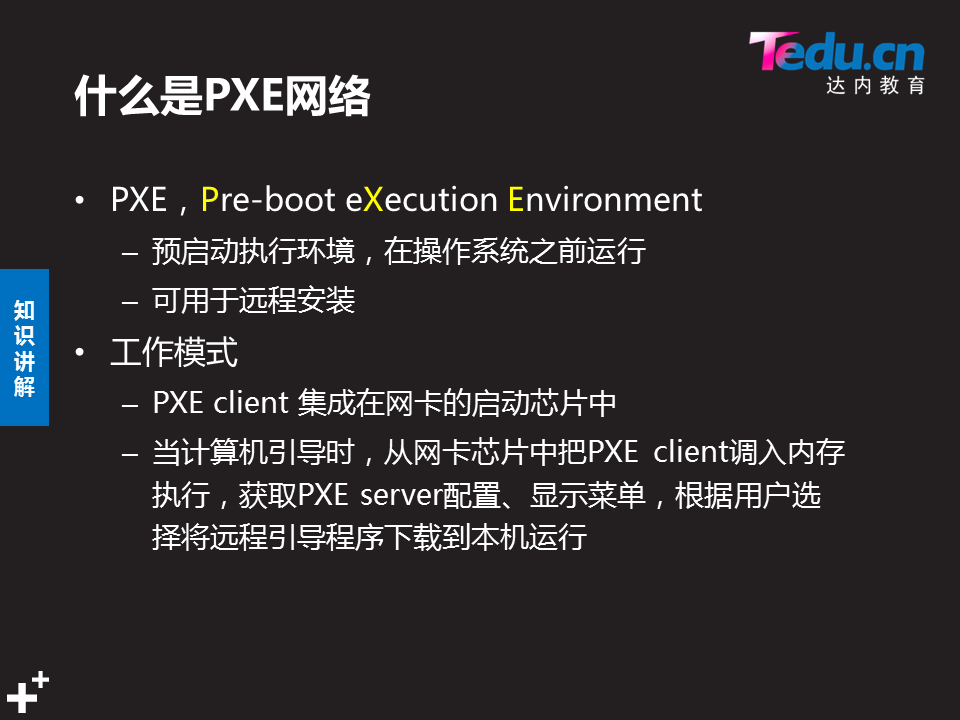
10 max-lease-time 7200;

11 }

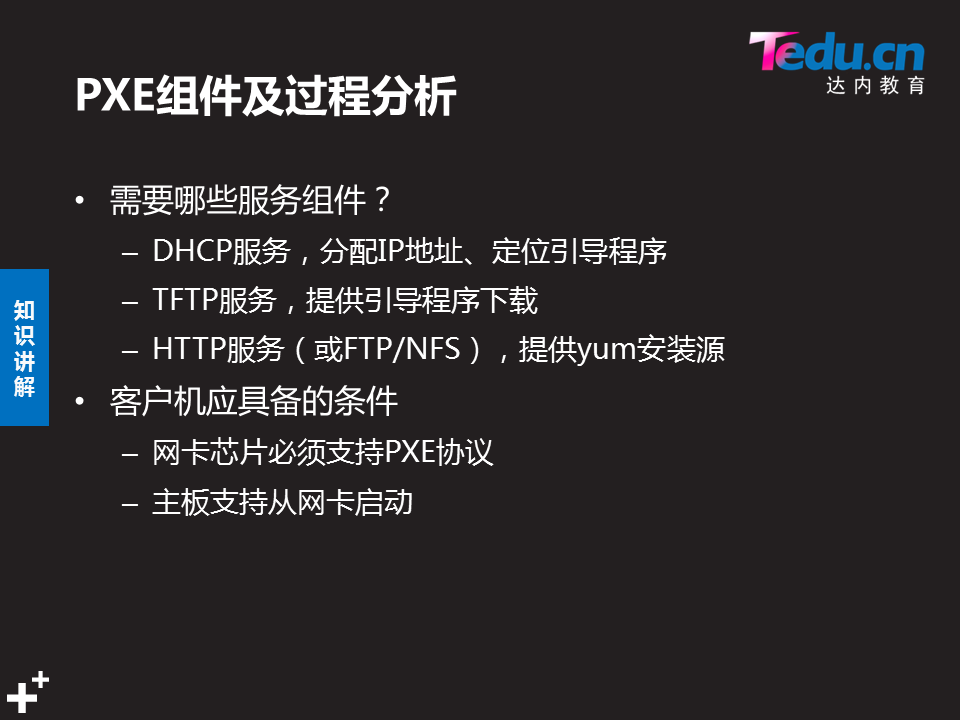
~

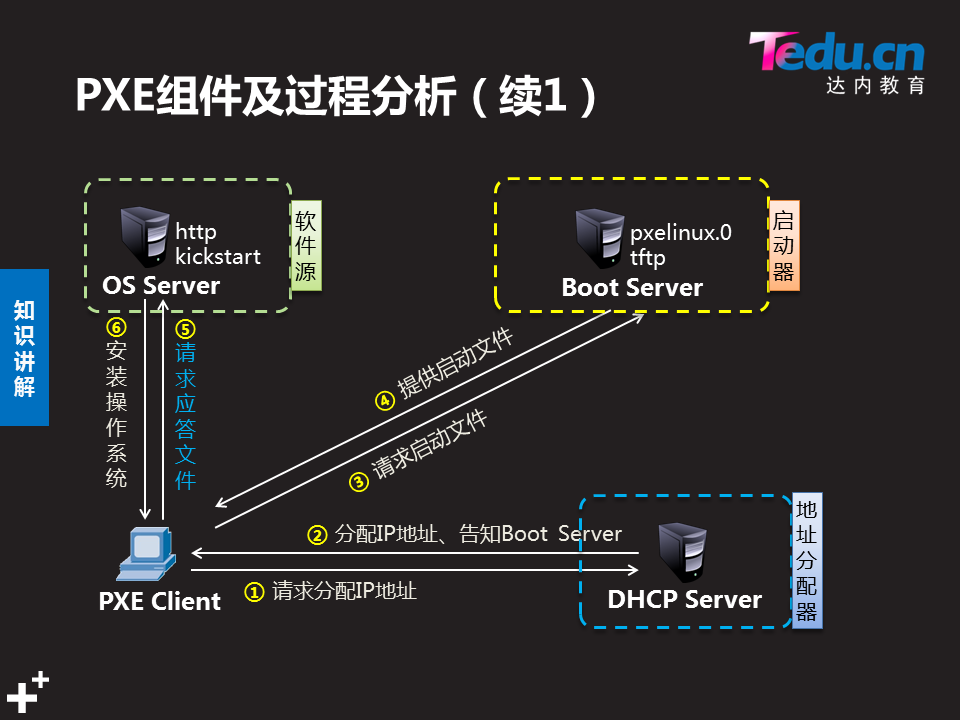
## PXE网络装机概述



客户端机器网卡必须有PXE client 软件

服务端



TFTP :简单的文件传输协议（不需要验证用户名和密码），端口：69

引导文件（众多的驱动，CPU 硬盘等）

HTTP： 提供众多的系统安装包

虚拟机A：部署网络装机服务器

## 一、部署DHCP

[root@svr7 ~]# vim /etc/dhcp/dhcpd.conf

1 #

2 # DHCP Server Configuration file.

3 # see /usr/share/doc/dhcp\*/dhcpd.conf.example

4 # see dhcpd.conf(5) man page

5 subnet 192.168.4.0 netmask 255.255.255.0 { ---修改

6 range 192.168.4.20 192.168.4.100; ---修改

7 option domain-name-servers 192.168.4.7; ---修改

8 option routers 192.168.4.254; ---修改

9 default-lease-time 600;

10 max-lease-time 7200;

11 next-server 192.168.4.7; -------新增 指定下一个服务器IP地址

12 filename "pxelinux.0"; ------新增指定网卡引导文件名称（告诉客户机，从下个服务 器获取这个pxelinux.0的文件）

13 }

[root@svr7 ~]# systemctl restart dhcpd

pxelinux.0:网卡引导文件，此文件为网络安装说明书，二进制文件，非常底层的文件，安装一个软件而默认产生以此命名的文件

su

## 二、搭建TFTP服务

tftp： 简单的文件传输协议 端口：69 默认共享路径：/var/lib/tftpboot

### 1.安装软件宝 tftp-setver

[root@svr7 ~]# yum -y install tftp-server

[root@svr7 ~]# systemctl restart tftp

[root@svr7 ~]# systemctl enable tftp

[root@svr7 ~]# ls /var/lib/tftpboot ---------------FTP的默认共享文件目录

### 2.部署pxelinux.0 文件

[root@svr7 ~]# yum provides /etc/passwd ---查询由哪个程序产生的/etc/passwd文件

[root@svr7 ~]# yum provides \*/pxelinux.0 ---查询所有的pxelinux.0 文件是由哪个程序产生的

已加载插件：langpacks, product-id, search-disabled-repos, subscription-manager

This system is not registered with an entitlement server. You can use subscription-manager to register.

rhel7/filelists\_db | 3.4 MB 00:00

s**yslinux-4.05-13.el7.x86\_64** : Simple kernel loader which boots from a FAT

: filesystem

源 ：rhel7

匹配来源：

文件名 ：/usr/share/syslinux/pxelinux.0

[root@svr7 ~]# yum -y install syslinux ----查找到之后，安装

[root@svr7 ~]# rpm -ql syslinux | grep pxelinux.0 ---查询安装清单

/usr/share/syslinux/gpxelinux.0

/usr/share/syslinux/pxelinux.0 ----找到此pxelinux.0 文件的所在目录

[root@svr7 ~]# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/ ---cp 到tftp共享目录里面

[root@svr7 ~]# ls /var/lib/tftpboot/ ------确定操作正确

pxelinux.0引导客户机操作：1.读取菜单文件/var/lib/tftpboot/pxelinux.cfg/default 需要创建

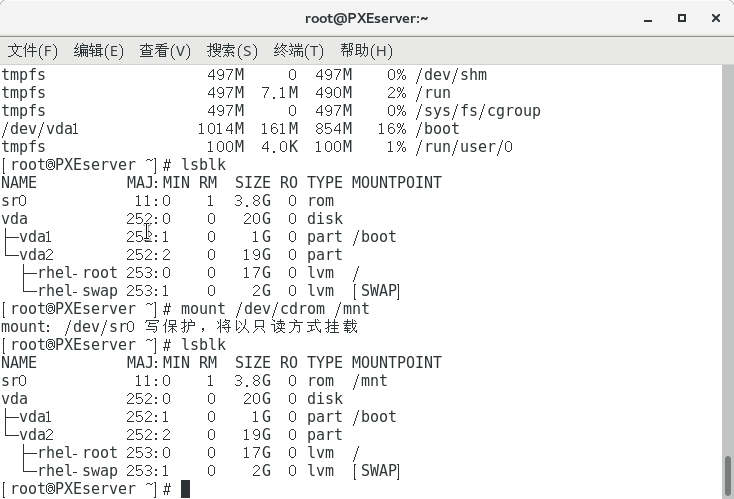
菜单文件如何来：1手写（很难写很难写）

2找模板，拷贝再修改

### 3部署菜单文件l

[root@svr7 ~]# mount /dev/cdrom /mnt

\mount: /dev/sr0 写保护，将以只读方式挂载/

[root@svr7 ~]# ls /mnt

[root@svr7 ~]# mkdir /var/lib/tftpboot/pxelinux.cfg/ ---创建 pxelinux.cfg目录（固定格式）

[r@svr7 ~]# cp /mnt/isolinux/isolinux.cfg /var/lib/tftpboot/pxelinux.cfg/default-拷贝菜单文件

pxelinux.cfg ：放菜单文件的目录

default ：菜单文件 来源 cp /mnt/isolinux/isolinux.cfg /var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 ~]# ls -l /var/lib/tftpboot/pxelinux.cfg/default

-r--r--r--. 1 root root 3166 12月 21 12:04 /var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 ~]# chmod u+w /var/lib/tftpboot/pxelinux.cfg/default --只是因为方便才修改权限

[root@svr7 ~]# ls -l /var/lib/tftpboot/pxelinux.cfg/default

### 4.部署图形模块

[root@svr7 ~]# vim /var/lib/tftpboot/pxelinux.cfg/default -----查看菜单文件内容

default vesamenu.c32 --加载菜单图形模块小软件

menu background splash.png 装机背景图—640\*480像素，大小50K左右，

[root@svr7 ~]# ls /mnt/isolinux/ -------查看光盘内的内容，如下

boot.cat grub.conf isolinux.bin memtest TRANS.TBL vesamenu.c32

boot.msg initrd.img isolinux.cfg splash.png upgrade.img vmlinuz

[root@svr7 ~]# cp /mnt/isolinux/vesamenu.c32 /mnt/isolinux/splash.png /var/lib/tftpboot/

[root@svr7 ~]# ls /var/lib/tftpboot/ -------查看此时tftp共享目录内的四个文件，如下

pxelinux.0 pxelinux.cfg splash.png vesamenu.c32

### 5.部署启动内核与启动所需的驱动程序

启动内核：只是管理启动，与操作系统内核无关，

启动所需的驱动程序：vmlinz包

[root@svr7 ~]# ls /mnt/isolinux/

boot.cat grub.conf isolinux.bin memtest TRANS.TBL vesamenu.c32

boot.msg initrd.img isolinux.cfg splash.png upgrade.img vmlinuz

[root@svr7 ~]# cp /mnt/isolinux/vmlinuz /mnt/isolinux/initrd.img /var/lib/tftpboot

[root@svr7 ~]# ls /var/lib/tftpboot ------查看此时tftp共享目录内的六个文件，如下

initrd.img pxelinux.0 pxelinux.cfg splash.png vesamenu.c32 vmlinuz

### 6.修改菜单文件

[root@svr7 ~]# vim /var/lib/tftpboot/pxelinux.cfg/default

1 default vesamenu.c32 ------加载图形模块

2 timeout 600 -----读秒时间1/10，一般修改60（6秒）

4 display boot.msg --------是否

10 menu background splash.png -----背景图片，可以更换

11 menu title Red Hat Enterprise Linux 7.4 ---标题:引导界面显示,改成‘NSD1811 PXE Server’

61 label linux ------每四行一个操作系统，多个操作系统要多个label,linux

62 menu label ^Install Red Hat Enterprise Linux 7.4 ---^I 是个快捷键，修改成Install RHEL7

62 menu label Install RHEL7 ------------上面 修改后菜单显示内容

63 menu default ------新增：读秒结束，则默认选择（第62行）如无此项将不会读秒

64 kernel vmlinuz --------加载内核

65 append initrd=initrd.img inst.stage2=hd:LABEL=RHEL-7.4\x20Server.x86\_64 quiet

65 append initrd=initrd.img 上面修改后

inst.stage2=hd:LABEL=RHEL-7.4\x20Server.x86\_64 quiet 光盘安装才需要此处PXE装机是删除

### 当前步骤总结：

一、DHCP---->ip地址 next-server filename

二、TFTP--→pxelinux.0

pxelinux.0 --->读取菜单文件/var/lib/tftpboot/pxelinux.cfg/default

default---->

1.pxelinux.0 名字：引导客户机操作 来源：安装syslinux

2.pxelinux.cfg ：放菜单文件的目录

default ：菜单文件 来源 cp /mnt/isolinux/isolinux.cfg /var/lib/tftpboot/pxelinux.cfg/default

3.initrd.img 图片 4.vmlinuz启动所需的驱动程序，是个安装包

5.splash.png 图片 6.vesamenu.c32 加载图形模块

第3-6个文件都是在系统镜像文件目录里面/mnt/isolinux，全部拷贝到 /var/lib/tftpboot/

[root@svr7~]# cp /mnt/isolinux/vesamenu.c32 /mnt/isolinux/splash.png /mnt/isolinux/vmlinuz /mnt/isolinux/initrd.img /var/lib/tftpboot/

[root@svr7~]# ls /var/lib/tftpboot/

initrd.img pxelinux.0 pxelinux.cfg splash.png vesamenu.c32 vmlinuz

[root@PXEserver isolinux]#

**验证：**

1新建一台虚拟机

安装方式选择PXE网络引导安装

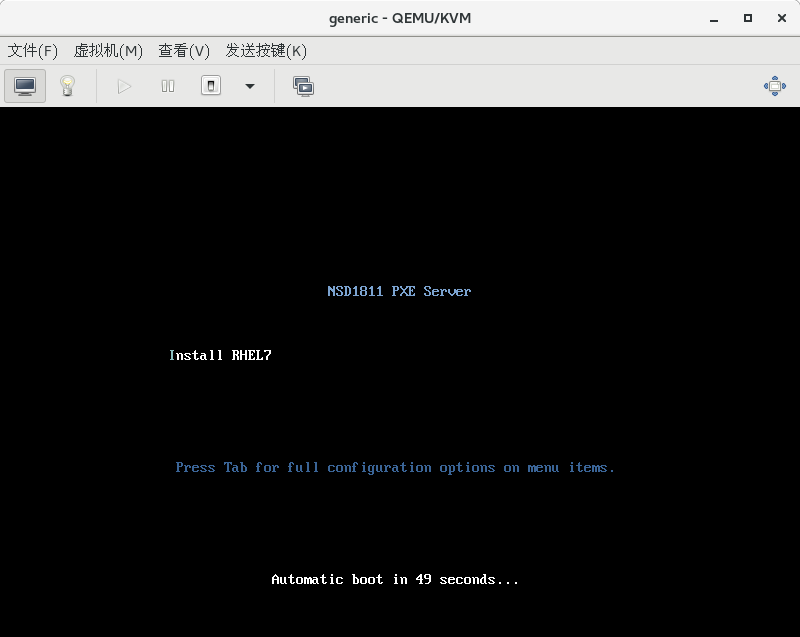
网络类型选择Private1 最终结果如下

NSD1811 PXE Server 名称是修改第11行

11 menu title Red Hat Enterprise Linux 7.4 ---标题:引导界面显示,改成‘NSD1811 PXE Server’

Install RHEL7 名称是修改第62行

62 menu label ^Install Red Hat Enterprise Linux 7.4 ---^I 是个快捷键，修改成Install RHEL7



## 三、搭建Web服务，共享光盘所有内容

1、安装httpd 软件，并启动服务

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

[root@svr7 ~]# ls /var/www/html

[root@svr7 ~]# mkdir /var/www/html/rhel7 ------新建目录，将用来挂载光盘

[root@svr7 ~]# mount /dev/cdrom /var/www/html/rhel7 ---挂载

mount: /dev/sr0 写保护，将以只读方式挂载

[root@svr7 ~]# ls /var/www/html/rhel7 -----查看是否挂载成功

addons extra\_files.json isolinux Packages RPM-GPG-KEY-redhat-release

EFI GPL LiveOS repodata TRANS.TBL

EULA images media.repo RPM-GPG-KEY-redhat-beta

[root@svr7 ~]# firefox 192.168.4.7/rhel7 -----访问测试

[root@svr7 ~]#

------此处挂载是临时的，现实中，应将光盘内容拷贝到web的目录内

------虚拟web主机的影响，如由虚拟WEB主机，访问firefox 192.168.4.7/rhel7 访问的将是虚拟主机第一个目录的/rhel7 目录

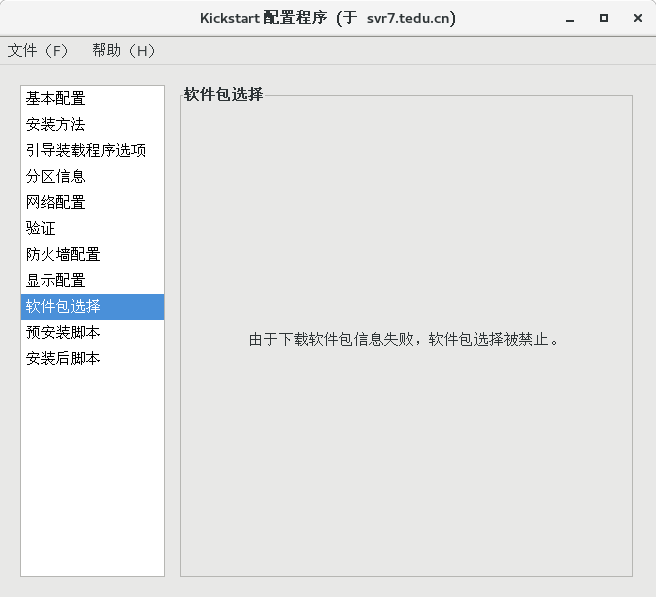
## 四、部署无人值守安装，生成应答文件

应答文件：可手写，可图形工具生成。回答如何分区，安装哪些包，选择什么时区、语言

1、图形工具system-config-kickstart生成应答文件

[root@svr7 ~]# yum -y install system-config-kickstart

[root@svr7 ~]# system-config-kickstart



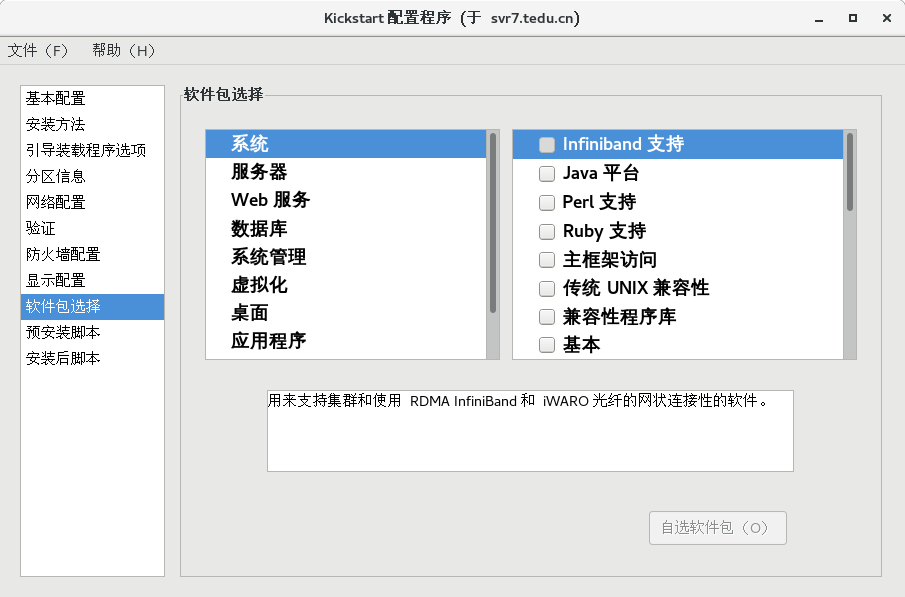
[root@svr7 ~]#system-config-kickstart

1.点击软件包选择，显示“由于下载软件包信息失败，软件包选择被禁止”，这个是rhel7的BUG ，rhel5/6是正常的，需要Yum仓库支持，必须要求本机光盘仓库标识为：[development]

如下：



[root@svr7 ~]# system-config-kickstart ---------------修改完成后再次启动



2填写分区信息—布局，

3.查看生成的ks.cfg 应答文件

[root@svr7 ~]# ls

1.txt initial-setup-ks.cfg 公共 视频 文档 音乐

anaconda-ks.cfg ks.cfg 模板 图片 下载 桌面

[root@svr7 ~]# vim ks.cfg

4.将应答文件放到httpd 共享目录内

[root@svr7 ~]# cp /root/ks.cfg /var/www/html/ -----将应答文件放到httpd 共享目录内

[root@svr7 ~]# ls /var/www/html/

ks.cfg rhel7

[root@svr7 ~]# firefox 192.168.4.7/ks.cfg

[Parent 6213] WARNING: pipe error: 断开的管道: file /builddir/build/BUILD/firefox-52.2.0/firefox-52.2.0esr/ipc/chromium/src/chrome/common/ipc\_channel\_posix.cc, line 685

[root@svr7 ~]#

5修改菜单文件，指定ks应答文件位置

[root@svr7 ~]# vim /var/lib/tftpboot/pxelinux.cfg/default

label linux

menu label ^Install RHEL7

menu default

kernel vmlinuz

append initrd=initrd.img ks=http://192.168.4.7/ks.cfg ---新增

## 五 第二次部署

(与前面步骤一样可能有所完善)

**一、部署DHCP**

[root@svr7 ~]# vim /etc/dhcp/dhcpd.conf

1 #

2 # DHCP Server Configuration file.

3 # see /usr/share/doc/dhcp\*/dhcpd.conf.example -----模板在此文件中：在末行模式中读 取此文件的内容到本文件并进行如下编辑:r /usr/share/doc/dhcp\*/dhcpd.conf.example

4 # see dhcpd.conf(5) man page

5 subnet 192.168.4.0 netmask 255.255.255.0 { ---修改：指定分配的网段

6 range 192.168.4.20 192.168.4.100; ---修改：指定分配地址范围

7 option domain-name-servers 192.168.4.7; ---修改：配置DNS

8 option routers 192.168.4.254; ---修改：配置的网关为

9 default-lease-time 600;

10 max-lease-time 7200;

11 next-server 192.168.4.7; -------新增 指定下一个服务器IP地址

12 filename "pxelinux.0"; ------新增：指定网卡引导文件名称（告诉客户机，从下个服 务器获取这个pxelinux.0的文件）

13 }

[root@svr7 ~]# systemctl restart dhcpd

**二、搭建TFTP服务**

tftp： 简单的文件传输协议 端口：69 默认共享路径：/var/lib/tftpboot

1.安装软件宝 tftp-setver

[root@svr7 ~]# yum -y install tftp-server

[root@svr7 ~]# systemctl restart tftp

[root@svr7 ~]# systemctl enable tftp

[root@svr7 ~]# ls /var/lib/tftpboot ---------------FTP的默认共享文件目录

**三、部署pxelinux.0 引导文件**

[root@svr7 ~]# yum provides /etc/passwd ---查询由哪个程序产生的/etc/passwd文件

[root@svr7 ~]# yum provides \*/pxelinux.0 ---查询所有的pxelinux.0 文件是由哪个程序产生的

已加载插件：langpacks, product-id, search-disabled-repos, subscription-manager

This system is not registered with an entitlement server. You can use subscription-manager to register.

rhel7/filelists\_db | 3.4 MB 00:00

s**yslinux-4.05-13.el7.x86\_64** : Simple kernel loader which boots from a FAT

: filesystem

源 ：rhel7

匹配来源：

文件名 ：/usr/share/syslinux/pxelinux.0

[root@svr7 ~]# yum -y install syslinux ----查找到程序之后，安装

[root@svr7 ~]# rpm -ql syslinux | grep pxelinux.0 ---查询安装清单找pxelinux.0 文件的目录

/usr/share/syslinux/gpxelinux.0

/usr/share/syslinux/pxelinux.0 ----找到此pxelinux.0 文件的所在目录

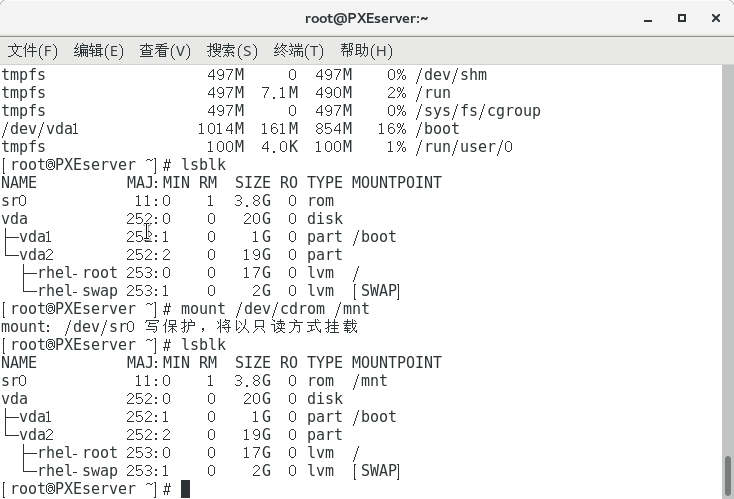
[root@svr7 ~]# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/ ---cp 到tftp共享目录里面

[root@svr7 ~]# ls /var/lib/tftpboot/ ------确定操作正确

**四、部署菜单文件**

[root@svr7 ~]# mount /dev/cdrom /mnt

\mount: /dev/sr0 写保护，将以只读方式挂载/

[root@svr7 ~]# ls /mnt

[root@svr7 ~]# mkdir /var/lib/tftpboot/pxelinux.cfg/ ---创建 pxelinux.cfg目录（固定格式）

[r@svr7 ~]# cp /mnt/isolinux/isolinux.cfg /var/lib/tftpboot/pxelinux.cfg/default-拷贝菜单文件

pxelinux.cfg ：放菜单文件的目录

default ：菜单文件 来源 cp /mnt/isolinux/isolinux.cfg /var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 ~]# ls -l /var/lib/tftpboot/pxelinux.cfg/default

-r--r--r--. 1 root root 3166 12月 21 12:04 /var/lib/tftpboot/pxelinux.cfg/default

[root@svr7 ~]# chmod u+w /var/lib/tftpboot/pxelinux.cfg/default --只是因为方便才修改权限

[root@svr7 ~]# ls -l /var/lib/tftpboot/pxelinux.cfg/default

第3-6个文件都是在系统镜像文件目录里面/mnt/isolinux，全部拷贝到 /var/lib/tftpboot/

[root@svr7~]# cp /mnt/isolinux/vesamenu.c32 /mnt/isolinux/splash.png /mnt/isolinux/vmlinuz /mnt/isolinux/initrd.img /var/lib/tftpboot/

[root@svr7~]# ls /var/lib/tftpboot/

initrd.img pxelinux.0 pxelinux.cfg splash.png vesamenu.c32 vmlinuz

[root@PXEserver isolinux]#

**六.修改菜单文件**

[root@svr7 ~]# vim /var/lib/tftpboot/pxelinux.cfg/default

1 default vesamenu.c32 ------加载图形模块

2 timeout 600 -----读秒时间1/10，一般修改60（6秒）

4 display boot.msg --------是否

10 menu background splash.png -----背景图片，可以更换

11 menu title Red Hat Enterprise Linux 7.4 ---标题:引导界面显示,改成‘NSD1811 PXE Server’

61 label linux ------每四行一个操作系统，多个操作系统要多个label,linux

62 menu label ^Install Red Hat Enterprise Linux 7.4 ---^I 是个快捷键，修改成Install RHEL7

62 menu label Install RHEL7 ------------上面 修改后菜单显示内容

63 menu default ------新增：读秒结束，则默认选择（第62行）如无此项将不会读秒

64 kernel vmlinuz --------加载内核

65 append initrd=initrd.img inst.stage2=hd:LABEL=RHEL-7.4\x20Server.x86\_64 quiet

65 append initrd=initrd.img 上面修改后

inst.stage2=hd:LABEL=RHEL-7.4\x20Server.x86\_64 quiet 光盘安装才需要此处PXE装机是删除

**验证：**

1新建一台虚拟机

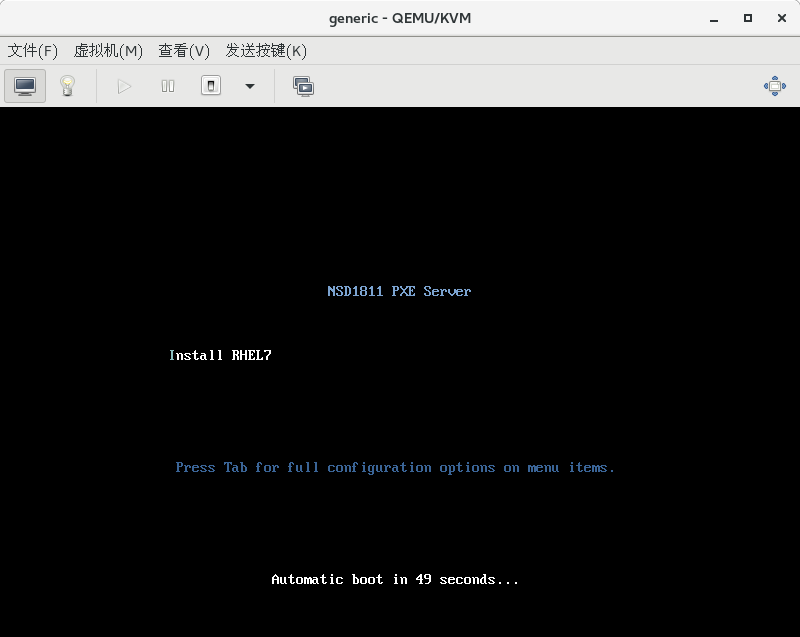
安装方式选择PXE网络引导安装 网络类型选择Private1 最终结果如下

NSD1811 PXE Server 名称是修改第11行

11 menu title Red Hat Enterprise Linux 7.4 ---标题:引导界面显示,改成‘NSD1811 PXE Server’

Install RHEL7 名称是修改第62行

62 menu label ^Install Red Hat Enterprise Linux 7.4 ---^I 是个快捷键，修改成Install RHEL7



**看到此处前面步骤安装成功**

**七、搭建Web服务，共享光盘所有内容**

安装httpd 软件，并启动服务

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

[root@svr7 ~]# ls /var/www/html

[root@svr7 ~]# mkdir /var/www/html/rhel7 ------新建目录，将用来挂载光盘

[root@svr7 ~]# mount /dev/cdrom /var/www/html/rhel7 ---挂载

mount: /dev/sr0 写保护，将以只读方式挂载

[root@svr7 ~]# ls /var/www/html/rhel7 -----查看是否挂载成功

addons extra\_files.json isolinux Packages RPM-GPG-KEY-redhat-releasels

EFI GPL LiveOS repodata TRANS.TBL

EULA images media.repo RPM-GPG-KEY-redhat-beta

[root@svr7 ~]# firefox 192.168.4.7/rhel7 -----访问测试rhel7是否正常

[root@svr7 ~]#

------此处挂载是临时的，现实中，应将光盘内容拷贝到web的目录内

------虚拟web主机的影响，如由虚拟WEB主机，访问firefox 192.168.4.7/rhel7 访问的将是虚拟主机第一个目录的/rhel7 目录

**八、部署无人值守安装，生成应答文件**

应答文件：可手写，可图形工具生成。回答如何分区，安装哪些包，选择什么时区、语言

1、图形工具system-config-kickstart生成应答文件

[root@svr7 ~]# yum -y install system-config-kickstart ----安装

[root@svr7 ~]# system-config-kickstart ----运行此图形工具

[root@svr7 ~]# system-config-kickstart

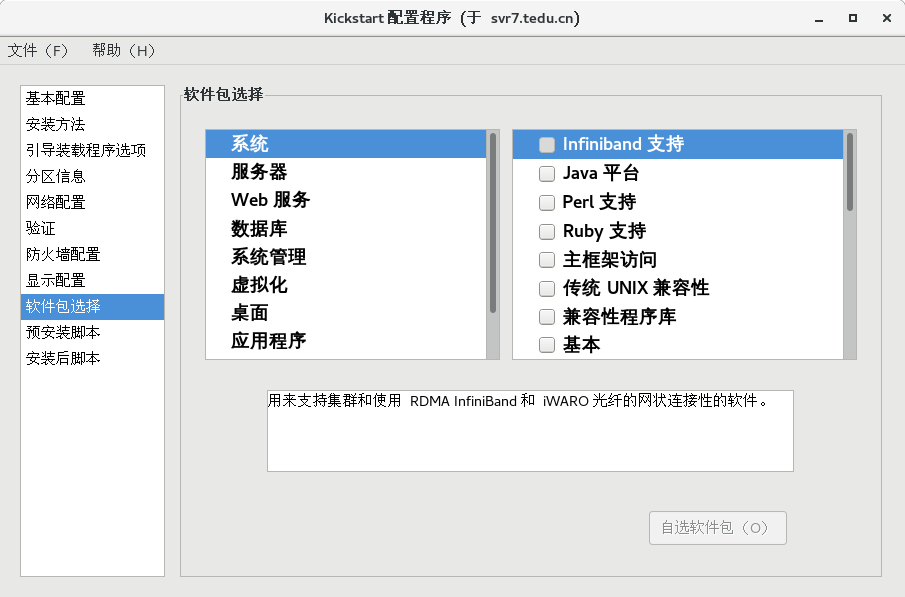
点击软件包选择，显示“由于下载软件包信息失败，软件包选择被禁止”，这个是rhel7的BUG ，rhel5/6是正常的，需要Yum仓库支持，必须要求本机光盘仓库标识为：[development]

如下：



[root@svr7 ~]# system-config-kickstart ---------------修改完成后再次启动

配置应答文件配置工具，生成ks.cfg 文件



查看生成的ks.cfg 应答文件

[root@svr7 ~]# ls

1.txt initial-setup-ks.cfg 公共 视频 文档 音乐

anaconda-ks.cfg ks.cfg 模板 图片 下载 桌面

[root@svr7 ~]# vim ks.cfg

[root@svr7 ~]# cp /root/ks.cfg /var/www/html/ -----将应答文件放到httpd 共享目录内

[root@svr7 ~]# ls /var/www/html/

ks.cfg rhel7

[root@svr7 ~]# firefox 192.168.4.7/ks.cfg ----测试访问ks.cfg是否正常

[Parent 6213] WARNING: pipe error: 断开的管道: file /builddir/build/BUILD/firefox-52.2.0/firefox-52.2.0esr/ipc/chromium/src/chrome/common/ipc\_channel\_posix.cc, line 685

[root@svr7 ~]#

[root@svr7 ~]# vim /var/lib/tftpboot/pxelinux.cfg/default 修改菜单文件指定ks应答文件位置

label linux

menu label ^Install RHEL7

menu default

kernel vmlinuz

append initrd=initrd.img ks=http://192.168.4.7/ks.cfg ---新增指定ks应答文件位置

开始 新建一台虚拟机

安装方式选择PXE网络引导安装 网络类型选择Private1 最终结果如下

# 系统与服务管理进阶SERVICES05

## 源码编译安装

源码要运行，必须先转成二进制的机器码。这是编译器的任务。

**什么是源码包软件；**

顾名思义，源码包就是源代码的可见的软件包，基于Linux和BSD系统的软件最常见

软件是否是开源软件，得具备两个条件；一是源代码可见；二是要有宽松的许可证书，比如GPL证书等；

**源代码有何用？**

一个软件的如果有源码，是任何人都能看到他是怎么开发而来的，就像一个瓶子，比如瓶子制作的模具是什么；需要什么材料；具体的用途以及瓶子的详细说明书等等。软件的开放源码就是类似，开发者在给我们软件的同时，也会告诉我们软件是怎么开发出来的；只要我们的水平足够的高，所有的代码都在那里，我们就可以修改和定制软件，以适合我们的需一.编译的具体过程

### 1. 配置（./configure）

编译器在开始工作之前，需要知道当前的系统环境，比如标准库在哪里、软件的安装位置在哪里、需要安装哪些组件等等。这是因为不同计算机的系统环境不一样，通过指定编译参数，编译器就可以灵活适应环境，编译出各种环境都能运行的机器码。这个确定编译参数的步骤，就叫做"配置"（configure）。

这些配置信息保存在一个配置文件之中，约定俗成是一个叫做configure的脚本文件。通常它是由autoconf工具生成的。编译器通过运行这个脚本，获知编译参数。

configure脚本已经尽量考虑到不同系统的差异，并且对各种编译参数给出了默认值。如果用户的系统环境比较特别，或者有一些特定的需求，就需要手动向configure脚本提供编译参数。

$ ./configure --prefix=/www --with-mysql

上面代码是php源码的一种编译配置，用户指定安装后的文件保存在www目录，并且编译时加入mysql模块的支持。要；如果Windows开放源代码，并以GPL发布，一样是有人能造出来N多的Windows发行版；遗憾的是Windows并不是开源系统；

### 2编译（make)

**头文件的预编译（precompilation）**

不同的源码文件，可能引用同一个头文件（比如stdio.h）。编译的时候，头文件也必须一起编译。为了节省时间，编译器会在编译源码之前，先编译头文件。这保证了头文件只需编译一次，不必每次用到的时候，都重新编译了。

不过，并不是头文件的所有内容，都会被预编译。用来声明宏的#define命令，就不会被预编译。

**预处理（Preprocessing）**

预编译完成后，编译器就开始替换掉源码中bash的头文件和宏。为了便于阅读，代码只截取了头文件中与源码相关的那部分，即fputs和FILE的声明，省略了stdio.h的其他部分（因为它们非常长）。另外，上面代码的头文件没有经过预编译，而实际上，插入源码的是预编译后的结果。编译器在这一步还会移除注释。

这一步称为"预处理"（Preprocessing），因为完成之后，就要开始真正的处理了。

**编译（Compilation）**

预处理之后，编译器就开始生成机器码。对于某些编译器来说，还存在一个中间步骤，会先把源码转为汇编码（assembly），然后再把汇编码转为机器码。

注：make (gcc), 其调用 gcc 执行编译的过程依赖于配置文件makefile

**连接（Linking）**

对象文件还不能运行，必须进一步转成��执行文件。如果你仔细看上一步的转码结果，会发现其中引用了stdout函数和fwrite函数。也就是说，程序要正常运行，除了上面的代码以外，还必须有stdout和fwrite这两个函数的代码，它们是由C语言的标准库提供的。

编译器的下一步工作，就是把外部函数的代码（通常是后缀名为.lib和.a的文件），添加到可执行文件中。这就叫做连接（linking）。这种通过拷贝，将外部函数库添加到可执行文件的方式，叫做静态连接（static linking），后文会提到还有动态连接（dynamic linking）。

make命令的作用，就是从第四步头文件预编译开始，一直到做完这一步。

### 3安装（make Install）

上一步的连接是在内存中进行的，即编译器在内存中生成了可执行文件。下一步，必须将可执行文件保存到用户事先指定的安装目录。

表面上，这一步很简单，就是将可执行文件（连带相关的数据文件）拷贝过去就行了。但是实际上，这一步还必须完成创建目录、保存文件、设置权限等步骤。这整个的保存过程就称为"安装"（Installation）。

**所以软件的源代码的用处无非是以下两点；**

1、软件根据用户的需要加以定制；

2、二次开发；注：要根据软件的许可证书约定为准，开发者许可二次开发才行；

1、准备安装包 ，

下面的方法二选一

[root@room9pc01 ~]# scp /root/桌面/tools.tar.gz [root@192.168.4.7](mailto:root@192.168.4.7):/root ----真机传送

root@192.168.4.66's password:

tools.tar.gz 100% 10MB 145.3MB/s 00:00

[root@A~]# wget <ftp://192.168.4.254/pub/tools.tar.gz> ----客户端下载

2、理论

源码包：最原始的安装包，灵活度高，可按需要选择与定制，有更多软件可供选择

软件是最新的及时修复了BUG

适用各种平台

源码包---》可执行软件---》运行安装

源码包用gcc 、make工具变成可执行软件

RPM包：由源码包编译、封装而来，不可修改 rpm -ivh 包名 yum -y install 包名

3操作

A： 解包

[root@room9pc01 ~]# scp /root/桌面/tools.tar.gz [root@192.168.4.7](mailto:root@192.168.4.7):/root --1下载包真机传送

[root@svr7 ~]# yum -y install gcc make ------安装gcc、make 两个工具

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

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

B：检测

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

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

[root@svr7 inotify-tools-3.13]# ./configure ---配置、指定安装目录/功能模块等模块，检测gcc是否安装

--disable-FEATURE -----不想要这个功能

--enable-shard ----开启这个功能

--prefix=指定安装路径

C：make 编译，生成可执行的二进制程序文件

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

D：安装，将编译好的文件复制到安装目录

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

[root@svr7 inotify-tools-3.13]# ls /usr/local/bin/inotifywait ---检测程序是否生成

/usr/local/bin/inotifywait

E:卸载, 我们可以找到软件的安装点删除。主要看你把它安装在哪了。

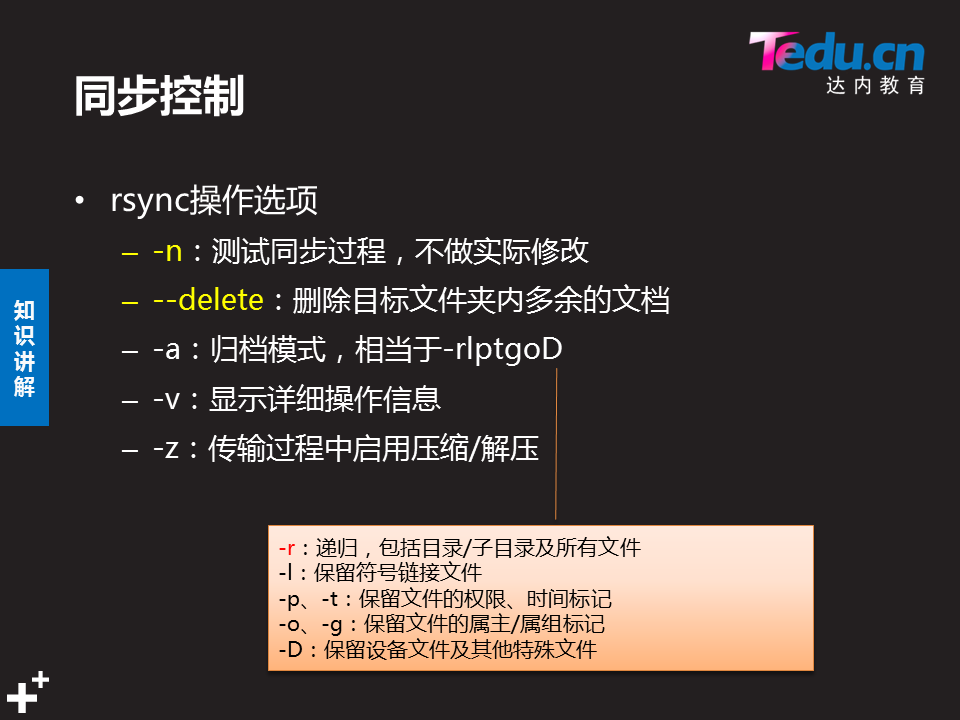
==========================================

## 二、rsync ----远程同步 Remte Sync

支持本地复制，或与其他SSH、rsync主机同步

同步与复制的差异：复制是完全拷贝源到目录，同步是增量拷贝，只传输变化过的数据

rsync [选项] 源目录 目标目录

选项 -a 一定会加上

注：源目录有没有带'/' 没有带斜线将会将整个目录同步，带斜线只同步目录下的所有数据

[root@svr7 ~]# mkdir /abc /dir ----创建 /abc /dir 准备看效果

[root@svr7 ~]# cp /etc/passwd /abc --/abc内放入passwd文件

[root@svr7 ~]# cp /etc/fstab /abc --/abc内放入fstab文件

[root@svr7 ~]# ls /abc /dir ---查看两个目录内的东西

/abc: ---/abc 内有fstab passwd 两个文件

fstab passwd

/dir: ---/dir 内没有文件

[root@svr7 ~]# rsync -avz /abc /dir ----同步文件，没有加“/”

sending incremental file list

abc/

abc/fstab

abc/passwd

sent 1217 bytes received 54 bytes 2542.00 bytes/sec

total size is 2524 speedup is 1.99

[root@svr7 ~]# ls /abc /dir ----同步之后再看两个目录的内容变化，没有加“/”

/abc:

fstab passwd

/dir:

abc -----没有加“/” 将abc 目录给弄过来了

[root@svr7 ~]# rm -rf /dir/\* ----晴空dir目录，

[root@svr7 ~]# rsync -avz /abc/ /dir ----再次同步，abc目录加“/”

[root@svr7 ~]# ls /abc /dir ----查看加“/”之后 将abc目录内的内容给弄过来了

[root@svr7 ~]# cp /etc/shadow /abc ---abc增加shadow 文件后

[root@svr7 ~]# rsync -avz /abc/ /dir 再次同步

[root@svr7 ~]# ls /abc /dir --查看dir目录多了 shadow 文件

/abc:

fstab passwd shadow

/dir:

fstab passwd shadow

-----------------------------------------------------------------------

[root@svr7 ~]# touch /dir/1.txt 目标目录多一个文件

[root@svr7 ~]# ls /dir /abc

/abc:

fstab passwd shadow

/dir:

1.txt fstab passwd shadow

[root@svr7 ~]# rsync -avz /abc/ /dir/

[root@svr7 ~]# ls /dir /abc

/abc:

fstab passwd shadow

/dir:

1.txt fstab passwd shadow

[root@svr7 ~]# rsync -avz --delete /abc/ /dir/ ---同步加上--delete 参数

sending incremental file list

deleting 1.txt

sent 78 bytes received 12 bytes 180.00 bytes/sec

total size is 3700 speedup is 41.11

[root@svr7 ~]# ls /abc /dir 同步加上--delete 参数后再查看内容

/abc:

fstab passwd shadow

/dir:

fstab passwd shadow ------1.txt 文件被删除

===============================================

### rsync+ssh 同步



[root@svr7 ~]# rsync -avz --deltete /opt/ root@192.168.4.207:/opt/

### 两台服务器之间的实时同步

服务器A 远程 B：验证优先级 公钥私钥>用户名与密码

服务器A生成公钥文件(锁)、私钥文件(钥匙)，将公钥传递给B

一、部署公钥私钥ssh无密码验证

[root@svr7 ~]# ssh-keygen -----------生成公钥私钥-一路回车

Generating public/private rsa key pair. -----------生成 public(公钥)/private(私钥)文件

Enter file in which to save the key (/root/.ssh/id\_rsa): -----------生成公钥私钥存放的路径

Enter passphrase (empty for no passphrase): ----设置远程密码

Enter same passphrase again: ----设置使用公钥私钥密码

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

id\_rsa id\_rsa.pub known\_hosts ----known\_hosts :记录了曾经远程过的服务器信息

二、传输公钥给远程客户端

[root@svr7 ~]# ssh-copy-id root@192.168.4.207 -----传输公钥给客户机

root@192.168.4.207's password: ----第一次传输还是要输入对方的用户登录密码

[root@svr7 ~]# ssh -X root@192.168.4.207 -----从svr7上登录，

Last login: Mon Dec 24 08:46:09 2018 from 192.168.4.254

[root@B ~]# ls /root/.ssh -----查看客户机上传输过来的公钥存放目录

authorized\_keys

[root@B ~]#

## 三、监控目录内容的变化

• Linux内核的 inotify 机制

– 提供事件响应式的文件系统通知机制

– 安装 inotify-tools 控制工具可调用此机制实现监控

attrib 权限的改变

[root@svr7 ~]# inotifywait -mrq /opt/

/opt/ OPEN,ISDIR

/opt/ CLOSE\_NOWRITE,CLOSE,ISDIR

/opt/ OPEN,ISDIR

/opt/ CLOSE\_NOWRITE,CLOSE,ISDIR

/opt/ DELETE 1.txt ----另外一个终端连接上svr7,删除了/opt/1.txt,被监控的内容

[root@svr7 /]# vim /root/rsync.sh ------创建一个脚本来实现监控与同步

#!/bin/bash

while inotifywait -rqq /opt/ ----判断监控opt目录是否执行成功，两个qq选项是无屏幕输出

do

rsync -az --delete /opt/ root@192.168.4.207:/opt/

done --------------------------------------可在此处加‘&’，使整个脚本后台运行

[root@svr7 /]# chmod +x /root/rsync.sh

[root@svr7 /]# /root/rsync.sh

[root@svr7 /]# /root/rsync.sh & ----‘&’放到后台运

## inotifywait工具

### 案例3:inotifywait监控使用

本例要求安装inotify-tools工具，并针对文件夹 /opt 启用 inotifywait 监控，完成下列任务：

当此目录下出现新建、修改、更改权限、删除文件等事件时能给出提示

验证上述监控事件的效果

3.2 方案

inotifywait监控操作：

inotifywait [选项] 目标文件夹

inotifywait常用命令选项：

-m，持续监控（捕获一个事件后不退出）

-r，递归监控、包括子目录及文件

-q，减少屏幕输出信息

-e，指定监视的 modify、move、create、delete、attrib 等事件类别

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装inotify-tools软件包

1. 解包

[root@inotifywaitweb tools]# yum -y install gcc make

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

1. 配置

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

[root@svr7 inotify-tools-3.13]# ./configure

3）编译

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

4）安装

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

5）检查安装结果（inotifywait程序可用）

[root@svr7 ~]# inotifywait --help

inotifywait 3.13

Wait for a particular event on a file or set of files.

Usage: inotifywait [ options ] file1 [ file2 ] [ file3 ] [ ... ]

Options:

-h|--help Show this help text.

.. ..

步骤二：测试inotifywait监控

1）开启监控任务，置入后台

[root@svr7 ~]# inotifywait -mrq -e create,modify,move,attrib,delete /opt &

[1] 55564

2）测试/opt/目录下的新建、修改、改名、更改权限、删除文件等事件的响应消息

观察新建文件时的监控信息：

[root@svr7 ~]# touch /opt/a.txt

/opt/ CREATE a.txt

/opt/ ATTRIB a.txt

观察修改文件内容时的监控信息：

[root@svr7 ~]# echo Hello > /opt/a.txt

[root@svr7 ~]# /opt/ MODIFY a.txt

/opt/ MODIFY a.txt

观察将文件改名时的监控信息：

[root@svr7 ~]# mv /opt/a.txt /opt/b.txt

/opt/ MOVED\_FROM a.txt

/opt/ MOVED\_TO b.txt

观察修改文件权限时的监控信息：

[root@svr7 ~]# chmod 600 /opt/b.txt

/opt/ ATTRIB b.txt

观察删除文件时的监控信息：

[root@svr7 ~]# rm -rf /opt/b.txt

/opt/ DELETE b.txt

3）停止监控任务

[root@svr7 ~]# kill -9 %1

[1]+ 已杀死 inotifywait -mr -e create,modify,move,attrib,delete /opt

[1] 30381

[root@svr7 /]# jobs #查看后台运行的脚本信息

### 案例4：配置Web镜像同步

4.1 问题

本例要求为两台Web服务器svr7、pc207的网页文档目录配置镜像同步，主要基于inotifywait监控技术实现实时触发操作，需要完成下列任务：

以 svr7 为发起方，原始目录为 /var/www/html/

以 pc207 为同步目标，基于SSH免密验证

编写 inotify+rsync 同步脚本，验证实时同步效果

4.2 方案

inotifywait与rsync的结合，主要思路：

while inotifywait监控操作

do

需要执行的rsync同步操作

done

4.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：为主机svr7、pc207部署同步目录

双方的目录均为/var/www/html/，如果安装了httpd，此目录会自动出现。

1）确认svr7的目录内容

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

.. ..

[root@svr7 ~]# ls /var/www/html/ //向目录下提供一些测试文件

libreoffice

2）确认pc207的目录内容

[root@pc207 ~]# yum -y install httpd

.. ..

[root@pc207 ~]# ls /var/www/html //初始目录无数据

[root@pc207 ~]#

步骤二：为svr7配置到pc207的SSH密钥对验证，实现免密码交互

1）检查当前用户是否已经有可用的SSH密钥对文件

[root@svr7 ~]# ls ~/.ssh/id\_\*

/root/.ssh/id\_rsa /root/.ssh/id\_rsa.pub

如果找不到id\_rsa、id\_rsa.pub密钥对文件，则需要执行下列操作创建：

[root@svr7 ~]# ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id\_rsa): //按回车，确认存放位置

Enter passphrase (empty for no passphrase): //按回车，确认不要密码

Enter same passphrase again: //再次按回车，确认

Your identification has been saved in /root/.ssh/id\_rsa.

Your public key has been saved in /root/.ssh/id\_rsa.pub.

The key fingerprint is:

00:a7:cb:2d:9d:b8:8a:df:f5:ff:5b:ed:bd:04:10:fe root@svr7

The key's randomart image is:

+--[ RSA 2048]----+

| . . . |

| + . . |

| . . o |

| . = o o |

| = + S E |

| o .. |

| . . ...|

| . o . . ....|

|..o . ....o. .+|

+-----------------+

2）将当前用户的SSH公钥部署到远程主机

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

The authenticity of host '192.168.4.207 (192.168.4.207)' can't be established.

ECDSA key fingerprint is d3:16:2c:9a:9d:91:28:c8:74:9c:af:2d:04:82:c9:66.

Are you sure you want to continue connecting (yes/no)? yes //首次连yes确认

root@192.168.4.207's password: //验证对方的密码

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@192.168.4.207'"

and check to make sure that only the key(s) you wanted were added.

3）验证免密码登录效果

[root@svr7 ~]# ssh root@192.168.4.207

Last login: Fri Jan 13 09:52:08 2017 from 192.168.4.110

[root@pc207 ~]# //确认已免密码连入远程主机

[root@pc207 ~]# exit //退出SSH登录环境

登出

Connection to 192.168.4.207 closed.

[root@svr7 ~]# //已反对原客户机

步骤三：编写镜像同步脚本并测试效果

1）编写脚本文件/root/isync.sh

[root@svr7 ~]# vim /root/isync.sh

#!/bin/bash

FROM\_DIR="/var/www/html/"

RSYNC\_CMD="rsync -az --delete $FROM\_DIR root@192.168.4.207:/var/www/html"

while inotifywait -rqq -e modify,move,create,delete,attrib $FROM\_DIR

do

$RSYNC\_CMD

done &

#两个qq

[root@svr7 ~]# chmod +x /root/isync.sh

2）运行脚本

[root@svr7 ~]# /root/isync.sh

[root@svr7 ~]# pgrep -l inotify //确认任务在运行

56494 inotifywait

3）测试同步效果

在svr7上向/var/www/html/目录下添加一个测试网页（触发同步）：

[root@svr7 ~]# touch /var/www/html/a.txt

[root@svr7 ~]# ls /var/www/html/

a.txt libreoffice

在pc207上检查/var/www/html/目录，内容应该已经与svr7上的同名目录一致：

[root@pc207 ~]# ls /var/www/html

a.txt libreoffice

4）结束测试后，在svr7上停止监控任务

[root@svr7 ~]# pkill -9 inotify

[root@svr7 ~]# pgrep -l inotify //确认已没有监控任务

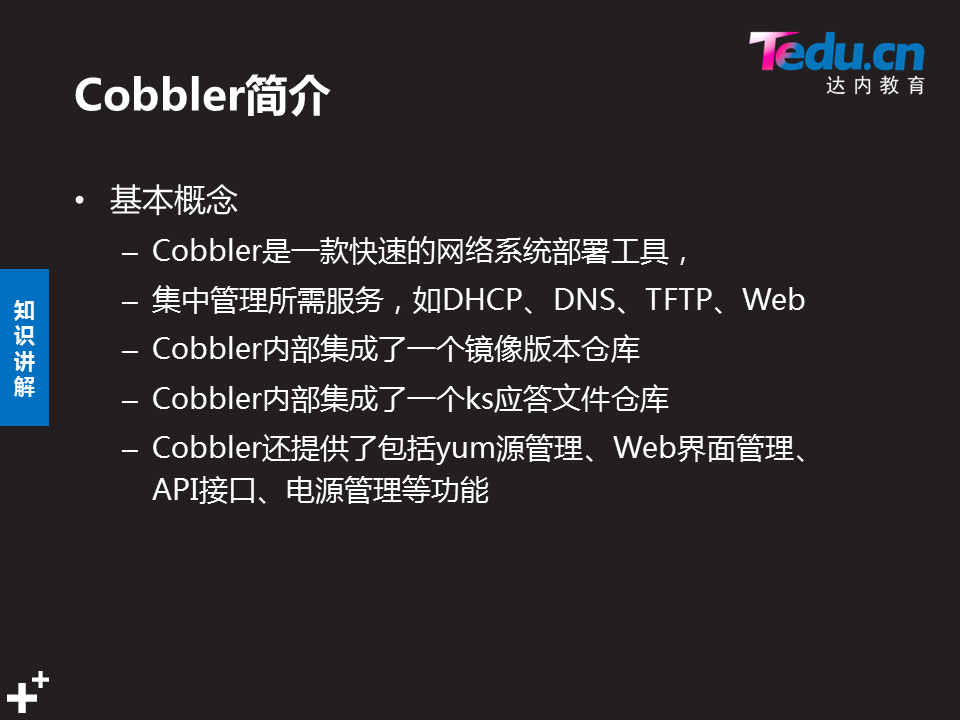
[root@svr7 ~]#]+ 运行中 /root/rsync.sh &

[root@svr7 /]# killall rsync.sh #杀死运行的脚本

====================================================================

## 四、Cobbler 装机平台

**效果，只要将光盘数据交给Cobbler，自动建立相应的应答文件和菜单文件，**



Ks 应答文件默认为最小化安装，

实施

1、准备一台CentOS 系统，

硬盘：50G 网络类型：private1 自动分区 安装包‘带GUI的服务器’

root 123456 lisi 123456

IP：192.168.4.100/24 主机名：Cobble.tedu.cn

搭建YUM仓库，设置防火墙为trusted、SElinux为宽松模式

[root@cobblier ~]# firewall-cmd --set-default-zone=trusted

success

[root@cobblier ~]# setenforce 0

[root@cobblier ~]# vim /etc/selinux/config

SELINUX=permissive

[root@cobblier ~]# unzip /root/Cobbler.zip -d / ---解压

[root@cobblier ~]# unzip /Cobbler/cobbler.zip -d /opt/ -------再次解压

[root@cobblier ~]# yum -y install /opt/cobbler/\*.rpm 使用YUM方式安装cobbler主程序、工具包，此方法可以解决一些以来关系

包介绍

cobbler.x86\_64 0:2.8.2-1.el7 cobbler-web.noarch 0:2.8.2-1.el7

2、配置cobler

[root@cobblier ~]# vim /etc/cobbler/settings

next\_server: 192.168.4.66 --设置下一个服务器还是为本机 注意空格

server: 192.168.4.66 ---设置本机为cobbler 服务器

manage\_dhcp: 1 ---设置cobbler管理DHCP服务

pxe\_just\_once: 1 ---防止客户端重复安装操作系统

开机启动项：匹配即停止

1、硬盘启动 2、光驱设备 3、U盘 4、网络引导

3、配置cobbler 的DHCP

[root@cobblier ~]# vim /etc/cobbler/dhcp.template

:%s /旧内容/新内容/g ---全文替换，在末行模式中

:%s /192.168.1/192.168.4/g

5 次替换，共 4 行

4、绝对路径解压 cobbler\_boot.tar.gz -----众多的引导文件

[root@cobblier ~]# tar -tf /Cobbler/cobbler\_boot.tar.gz --查看压缩包

tar: 从成员名中删除开头的“/”

/var/lib/cobbler/loaders/

/var/lib/cobbler/loaders/README

/var/lib/cobbler/loaders/COPYING.elilo

/var/lib/cobbler/loaders/COPYING.yaboot

/var/lib/cobbler/loaders/COPYING.syslinux

/var/lib/cobbler/loaders/elilo-ia64.efi

/var/lib/cobbler/loaders/yaboot

/var/lib/cobbler/loaders/pxelinux.0 -----这个文件按照此目录存放

/var/lib/cobbler/loaders/menu.c32

/var/lib/cobbler/loaders/grub-x86.efi

/var/lib/cobbler/loaders/grub-x86\_64.efi

----------/var/lib/cobbler/loaders/ 是安装cobbler的默认路径

[root@cobblier ~]# tar -xPf /Cobbler/cobbler\_boot.tar.gz –--按绝对路径释放包‘-P’(大写)选项

5、启动相关服务 cobblerd httpd tftp rsyncd

[root@cobblier ~]# systemctl restart cobblerd

[root@cobblier ~]# systemctl enable cobblerd

[root@cobblier ~]# systemctl restart httpd

[root@cobblier ~]# systemctl enable httpd

[root@cobblier ~]# systemctl restart tftp

[root@cobblier ~]# systemctl enable tftp

[root@cobblier ~]# systemctl restart rsyncd

[root@cobblier ~]# systemctl enable rsyncd

[root@cobblier ~]#

可使用for 循环启动

[root@cobblier ~]# for I in cobblerd httpd tftp rsyncd

>do

>

6、同步刷新cobbler 配置

[root@cobblier ~]# cobbler sync ----刷新、检查cobbler 是否配置成功

\*\*\* TASK COMPLETE \*\*\* -----最后显示 TASK COMPLETE 则顺利完成

[root@cobblier ~]# firefox <https://192.168.4.66/cobbler_web> ---访问网站

user：cobbler passwd：cobbler

.排错

配置了settings dhcp.template 两个配置文件

[root@cobblier ~]# vim /etc/cobbler/settings

[root@cobblier ~]# vim /etc/cobbler/dhcp.template

如查询不到错误，可重新生成文件

[root@svr7 ~]# yum provides \*/settings ---查询所有的settings 文件是由哪个程序产生的

cobbler-2.8.2-1.el7.x86\_64 : Boot server configurator

源 ：installed

匹配来源：

文件名 ：/etc/cobbler/settings –---有很多项，可看到此目录跟配置文件一致， cobbler

[root@svr7 ~]#rm /etc/cobbler/settings --------删除settings 配置文件

[root@svr7 ~]#yum -y reinstall cobbler -------重新安装，重新生成配置文件settings，

[root@svr7 ~]#vim /etc/cobbler/settings –-----再去修改并重启动相关服务

cobbler应用

[root@cobblier ~]# mount /dev/cdrom /dvd ------此时虚拟机挂载的是CentOS7系统

[root@cobblier ~]# cobbler import --path=/dvd –name=CentOS7

---从/dvd目录里面导入系统到/var/lib/cobbler/kickstarts/ 并命名为：CentOS7 需要时间

\*\*\* TASK COMPLETE \*\*\* ----成功

cobbler import –path=挂载点 –name=导入系统名称

cobbler将挂载之后的/dvd挂载了的系统，镜像导入到了/var/www/cobbler/ks\_mirror目录

/var/lib/cobbler/kickstarts/

[root@cobblier ~]# cobbler list --------查看cobbler 已有系统列表

distros:

CentOS7-x86\_64

profiles:

CentOS7-x86\_64

新建虚拟机，选用PXE装机方式，内存必须至少要2G

新系统的用户名为：root 密码：cobbler

[root@cobblier ~]# umount /dvd

[root@cobblier ~]# mount /dev/cdrom /dvd ---更换了光盘内容换了rhel7

mount: /dev/sr0 写保护，将以只读方式挂载

[root@cobblier ~]# cobbler import --path=/dvd --name=rhel7

---再次从/dvd目录里面导入系统到/var/lib/cobbler/kickstarts/ 并命名为：CentOS7 需要时间

[root@cobblier ~]# cobbler list ----再次查看cobbler已有系统列表多了rhel7-x86\_64

distros:

CentOS7-x86\_64

rhel7-x86\_64

profiles:

CentOS7-x86\_64

rhel7-x86\_64

新建虚拟机，选用PXE装机方式，内存必须至少要2G

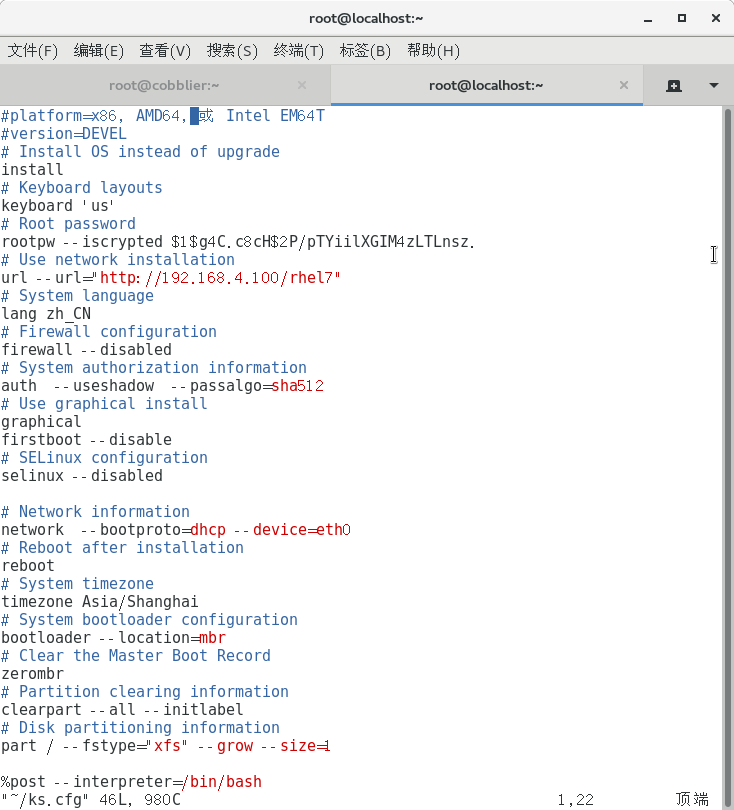
**自定义应答文件：开头注释行删除**

[root@cobbler ~]# system-config-kickstart #生成ks文件

必须默认kickstart文件存放位置：/var/lib/cobbler/kickstarts/

[root@cobbler ~]# cobbler list

修改kickstart文件：



[root@cobbler ~]# cobbler profile edit --name=CentOS7-x86\_64 --kickstart=/var/lib/cobbler/kickstarts/自定义.cfg

[root@cobbler ~]# cobbler profile report

[root@cobbler ~]# cobbler sync #同步配置

# 系统与服务管理进阶SERVICES06

## shell

在计算机科学中，Shell俗称壳（用来区别于核），是指“为使用者提供操作界面”的软件（命令解析器）。它类似于DOS下的command.com和后来的cmd.exe。它接收用户命令，然后调用相应的应用程序。

**基本上shell分两大类：**

一：图形界面shell（Graphical User Interface shell 即 GUI shell）

例如：应用最为广泛的 Windows Explorer （微软的windows系列操作系统），还有也包括广为人知的 Linux shell，其中linux shell 包括 X window manager (BlackBox和FluxBox），以及功能更强大的CDE、GNOME、KDE、 XFCE。

二：命令行式shell（Command Line Interface shell ，即CLI shell）

例如：bash / sh / ksh / csh（Unix/linux 系统） （MS-DOS系统）

cmd.exe/ 命令提示字符（Windows NT 系统）

Windows PowerShell（支持 .NET Framework 技术的 Windows NT 系统）

传统意义上的shell指的是命令行式的shell，以后如果不特别注明，shell是指命令行式的shell。

文字操作系统与外部最主要的接口就叫做shell。shell是操作系统最外面的一层。shell管理你与操作系统之间的交互：等待你输入，向操作系统解释你的输入，并且处理各种各样的操作系统的输出结果。

shell提供了你与操作系统之间通讯的方式。这种通讯可以以交互方式（从键盘输入，并且可以立即得到响应），或者以shell script(非交互）方式执行。shell script是放在文件中的一串shell和操作系统命令，它们可以被重复使用。本质上，shell script是命令行命令简单的组合到一个文件里面。

Shell基本上是一个命令解释器，类似于DOS下的command。它接收用户命令（如ls等），然后调用相应的应用程序。较为通用的shell有标准的Bourne shell (sh）和C shell (csh）。

交互式shell和非交互式shell

交互式模式就是shell等待你的输入，并且执行你提交的命令。这种模式被称作交互式是因为shell与用户进行交互。这种模式也是大多数用户非常熟悉的：登录、执行一些命令、签退。当你签退后，shell也终止了。

shell也可以运行在另外一种模式：非交互式模式。在这种模式下，shell不与你进行交互，而是读取存放在文件中的命令，并且执行它们。当它读到文件的结尾，shell也就终止了。

## **一 环境变量**

在Linux中,很多程序和脚本都通过环境变量来获取系统信息，存储临时数据，配置信息。环境变量是指用来存储有关shell会话和工作环境信息，允许你在内存中存储数据，以便程序或shell中运行的脚本能够轻松访问到它们。也是存储持久数据的一种简便方法。在bash shell中，环境变量分为：全局变量 局部变量 自定义变量。

默认情况下, bash shell会用一些特定的环境变量来定义系统的环境。

USER=当前系统登录的用户名

PATH=储存命令程序的搜寻路径，

[root@room9pc01 ~]# echo $PATH

/usr/lib64/qt3.3/bin: /usr/local/bin: /usr/local/sbin: /usr/bin:

/usr/sbin: /bin: /sbin: /root/bin

[root@svr7 ~]# vim /opt/hello.sh

[root@svr7 ~]# chmod +x /opt/hello.sh

[root@svr7 ~]# heello.sh

bash: heello.sh: 未找到命令...

[root@svr7 ~]# cd /opt

[root@svr7 opt]# heello.sh

bash: heello.sh: 未找到命令...

[root@svr7 opt]#

[root@svr7 opt]# cp hello.sh /usr/bin

[root@svr7 opt]# hello.sh

hallo

书写周期性计划任务，命令写绝对路径

### 全局环境变量

全局环境变量对于shell会话和所有生成的子shell都是可见的。局部变量则只是创建它们的shell可见。在linux系统中有系统环境变量，它们基本都是使用全大写字母，以区别普通用户的环境变量。要查看系统全局变量，可以使用env命令,



#### 查看全局变量

[root@rsync ~]# echo $USER

root

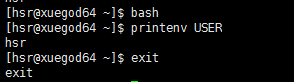
[root@rsync ~]# printenv USER

root

[root@rsync ~]# printenv PATH

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

正如前面所说的，全局变量在所有生成的子shell都是可见的，下面是用子shell输出USER全局变量，如下所示：



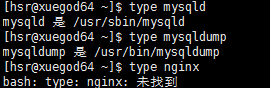
#### PATH环境变量

上面讲到了PATH变量，它的作用是：当你在shell命令行中输入一个外部命令时。Shell必须搜索系统来找到对应的程序。 PATH环境变量定义了用于进行命令和程序查找的目录。下面列出了要查找的5个目录。如果命令或者程序的位置没有包括在PATH变量中，那就必须使用绝对路径来查找，否则shell是无法找到的。找不到，它会产生一个错误信息。

[root@rsync ~]# echo $PATH

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

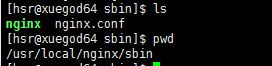
如下图所示：mysqld服务之所以能找到，是因为它存在于/usr/sbin目录下。Mysql的mssqldump工具包之所以能找到，是因为它存在于/usr/bin目录下。这二个目录都是存在于PATH全局变量中。而nginx服务命令却未找到。



　在重新加载nginx配置文件时，会提示未找到命令

180923155283835

　　下图来查找下nginx 服务是存放于哪个目录下。通过下图知道，nginx服务存在于/usr/local/nginx/sbin 目录下。而PATH环境变量中没有该路径。所示“提示未找到命令…“。



　　如果想在虚拟目录任意位置操作nginx服务,又不想每次都用cd绝对路径定位。有二种方法：1是前面章节所学的软链接，把nginx的绝对路径能过软链接指定到PATH环境下的一个目录下。2是将nginx绝对路径添加到PATH目录中。

**添加PATH环境变量**

　　上面讲到了nginx服务， 下面演示将nginx服务的绝对路径，添加到PATH环境变量中。如下图所示：已将nginxx服务路径添加到了PATH环境变量中。

180923155283837

　　下面来试试，在任意的虚拟目录下，执行nginx服务。下面是随意在虚拟目录的根目录下查看nginx服务， 重新加载nginx服务，该命令也找到了，下图的错误是nginx进程没有启动，与要介绍的PATH没有关系。

180923155283838

　　 总结: 当初学linux的时候，每当安装了一个服务时，在终端敲 bash执行该命令，经常提示未找到该命令。但有些服务又可以直接找到。现在明白了, shell查找命令会去指定的PATH环境目录下查找。当有些服务安装后，该服务的bin目录并没有在PATH环境目录中。

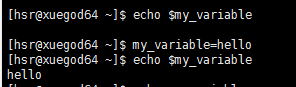
### 局部环境变量

局部环境变量虽然是局部的，但和全局环境变量一样重要。在linux操作系统上有默认全局环境变量也有局部环境变量，用户也可以自定义局部变量。

使用set 命令会显示某个指定进程下的所有环境变量，包括全局，局变，用户自定义全局变量。

### 自定义局部变量

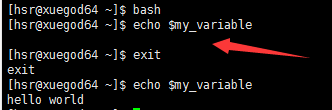
　　下是自定义一个局部变量，使用echo来声明一个变量名my\_variable，并赋值一个字符串(也可是数字)。然后再使用echo输出该变量的值。如下图所示：



　　要注意的点是:声明变量名时必须加$关键词, 赋值时不要加$关键词,输出时要加$关键词，也可以说使用echo时都要加$。变量名，等号，值三者之间不能有空格。如果变量值有空格时，必须加双引号。如下所示：

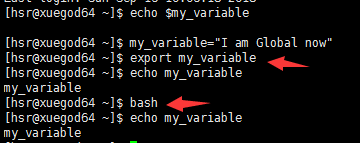
180923155081925

　　如果是在子shell中，那该变量就不存在，退回主线程还是存在，如下图所示：

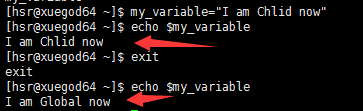


### 自定义全局变量

设定了全局变量，在该进程所创建的子进程中，该变量都是可见的。创建全局变量的方法是先创建一个局部变量,然后导出到全局环境中。通过export来导出，变量名前面不加$。如下图所示：

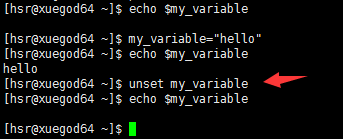


　　在子shell中修改全局变量并不会影响到父shell中该变量的值。这种改变仅在子shell中有效。并不会反映到父shell中。如下图所示：



### 删除环境变量

　　　　要删除环境变量可以用unset命令，在unset引用变量名时，不要加$。



## 二 下列yum源的配置文件配置哪里有误？

# vim /etc/yum.repos.d/abc.rope

[abc]

name=redhat

baseurl=ftp:///192.168.4.254/rhel7

enabled=0

gpgcheck=1

gpgcheck=1就必须加gpgkey=目录

gpgkey=目录 在安装光盘内

## **三 .权限掩码**umask

root用户创建的目录，默认权限为什么权限为755？其原因？

答案：默认创建的目录权限是drwxr-xr-x ,

权限掩码，umask

[root@svr7 opt]# umask

0022

777-022=755

[root@svr7 opt]# umask 077

[root@svr7 opt]# mkdir test

[root@svr7 opt]# ls -ld test

drwx------. 2 root root 6 12月 25 09:35 test 创建的文档默认权限为700

777-077=700

[root@svr7 opt]# umask 022

[root@svr7 opt]# mkdir test22

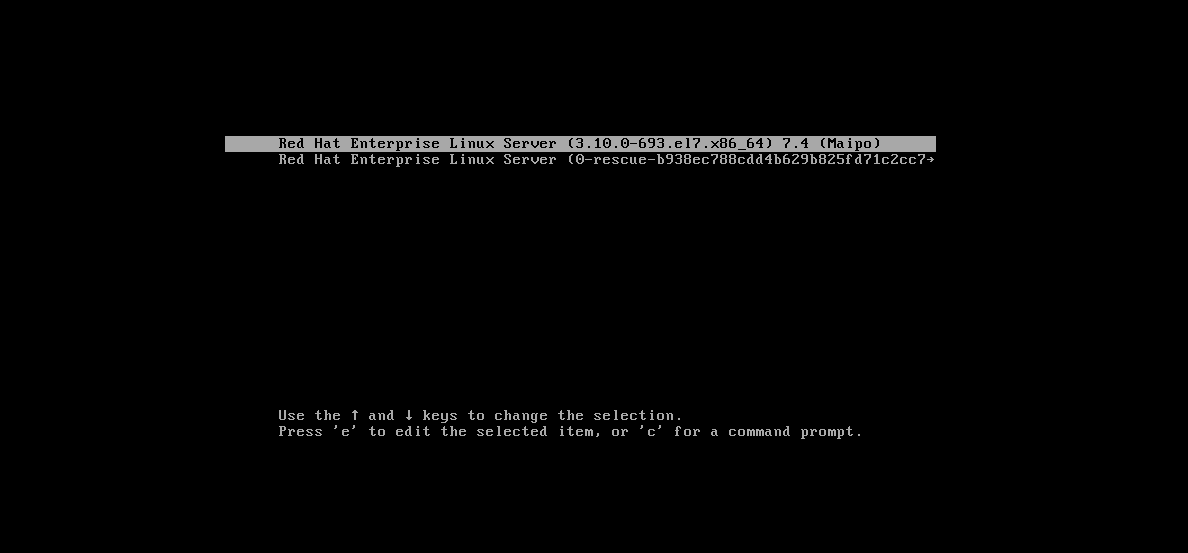
[root@svr7 opt]# ls -ld test22

drwxr-xr-x. 2 root root 6 12月 25 09:37 test22 创建的文档默认权限为755

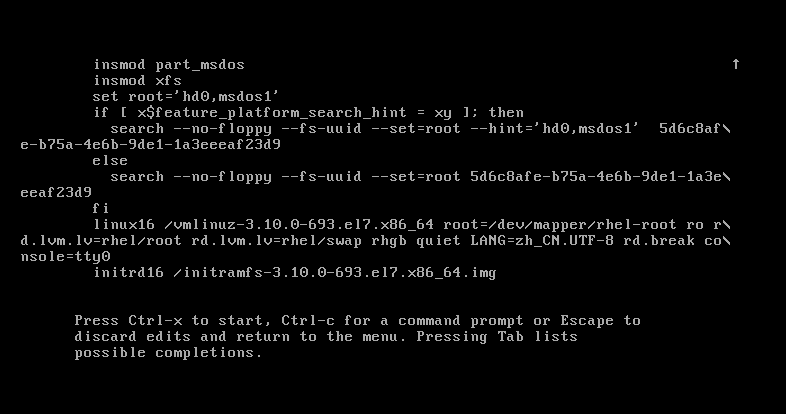
777-022=755

# CentOS7忘记密码重置密码

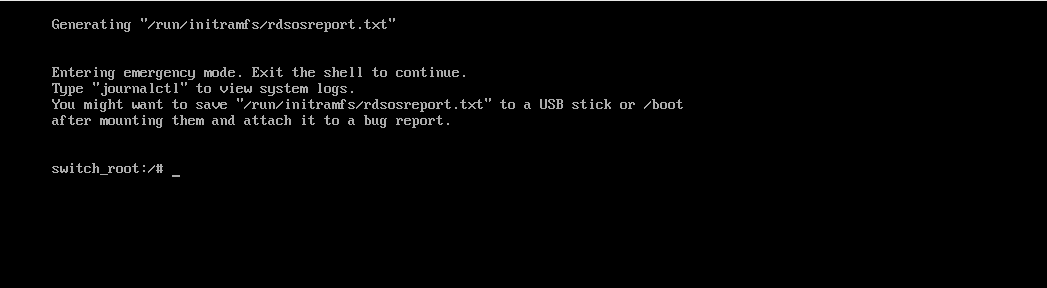
1,在GRUB菜单入下按e进入编辑状态



2找到linux16所在行,末尾添加rd.break console=tty0



3编辑好之后,按Ctrl+x 键进入恢复模式



4执行以下命令重设密码

switch\_root:/# mount -o remount,rw /sysroot #//以可读写方式重新挂载根系统

switch\_root:/# chroot /sysroot/ #//切换到根系统

sh-4.2# passwd root #//设置root用户的密码

#这里连续输入两次新密码

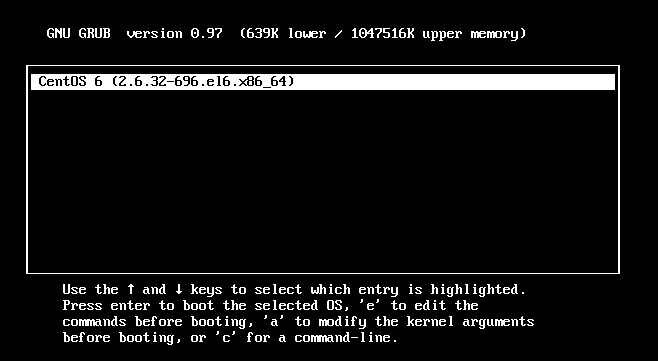
sh-4.2# touch /.autorelabel #//标记下一次启动重做 SELinux 标签

sh-4.2# exit

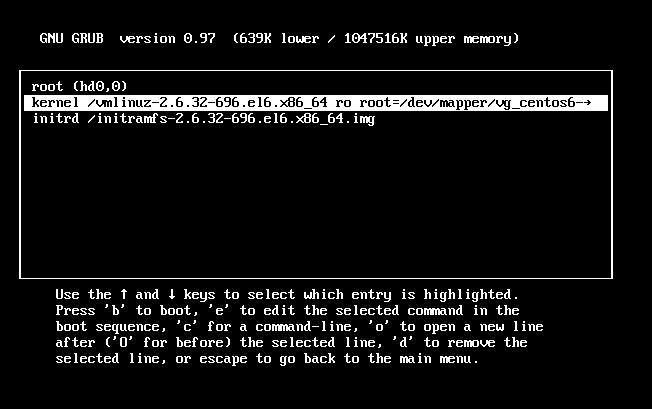
switch\_root:/# reboot #重启

# CentOS6忘记密码重置密码

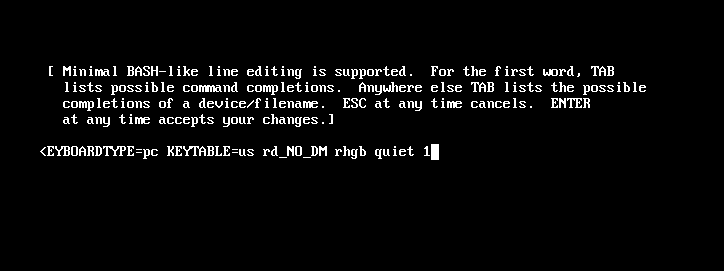
1.在开机启动的时候快速按键盘上的“E”键 或者“ESC”键（如果做不到精准快速可以在启动前一直按着或者不停的按），会进入如下界面。如果你的有多个操作系统就会出现多个内核，就会出现多个选项



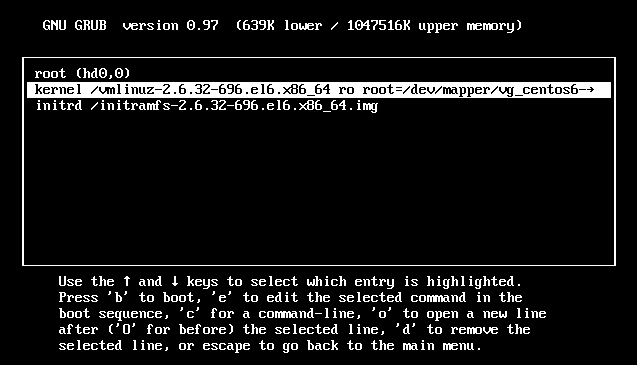
按E进入如下



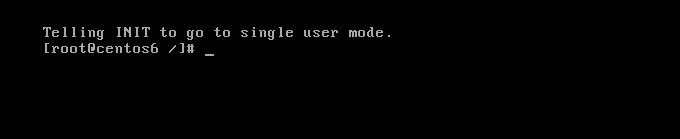
2选择第二项,”kernel”开头的,再次按E进入编辑, 空格输入 ”1”如下图,回车



3回车后将返回上一步,

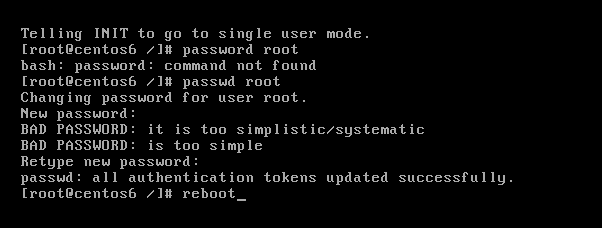


4,上图界面中按b 进入引导界面,如下图



5,运行命令

~]# passwd root #即修改root密码, 修改完rebbot重启即可



# Curl 的使用详解

<https://www.jianshu.com/p/07c4dddae43a> 参考网站

-I/--head只显示传输文档，经常用于测试连接本身

-o/--output把输出写到该文件中，必须输入保存文件名

-O/--remote-name把输出写到该文件中，保留远程文件的文件名

-F/--form模拟表单提交

-s/--silent静默模式，不输出任何东西

-S/--show-error显示错误，在选项 -s 中，当 curl 出现错误时将显示

-L/--location跟踪重定向

-f/--fai不输出错误

-n/--netrc从netrc文件中读取用户名和密码

--netrc-optional使用 .netrc 或者 URL来覆盖-n

--ntlm使用 HTTP NTLM 身份验证

-N/--no-buffer禁用缓冲输出

-p/--proxytunnel使用HTTP代理

--proxy-anyauth选择任一代理身份验证方法

--proxy-basic在代理上使用基本身份验证

--proxy-digest在代理上使用数字身份验证

--proxy-ntlm在代理上使用ntlm身份验证

-P/--ftp-port使用端口地址，而不是使用PASV

-M/--manual显示全手动

-Q/--quote文件传输前，发送命令到服务器

-r/--range检索来自HTTP/1.1或FTP服务器字节范围

--range-file读取（SSL）的随机文件

-R/--remote-time在本地生成文件时，保留远程文件时间

--retry传输出现问题时，重试的次数

--retry-delay传输出现问题时，设置重试间隔时间

--retry-max-time传输出现问题时，设置最大重试时间

--socks4用socks4代理给定主机和端口

--socks5用socks5代理给定主机和端口

-t/--telnet-option Telnet选项设置

--trace对指定文件进行debug

--trace-ascii Like跟踪但没有hex输出

--trace-time 跟踪/ 详细输出时，添加时间戳

-T/--upload-file 上传文件

-u/--user设置服务器的用户和密码

-U/--proxy-user 设置代理用户名和密码

-V/--version显示版本信息

-w/--write-out [format] 什么输出完成后

-x/--proxy 在给定的端口上使用HTTP代理

-X/--request 指定什么命令

-y/--speed-time放弃限速所要的时间。默认为30

-Y/--speed-limit停止传输速度的限制，速度时间'秒

-z/--time-cond传送时间设置

-0/--http1.0使用HTTP 1.0

-1/--tlsv1使用TLSv1（SSL）

-2/--sslv2使用SSLv2的（SSL）

-3/--sslv3使用的SSLv3（SSL）

--3p-quote like -Q for the source URL for 3rd party transfer

--3p-url 使用url，进行第三方传送

--3p-user 使用用户名和密码，进行第三方传送

-4/--ipv4 使用IP4

-6/--ipv6 使用IP6

-#/--progress-bar 用进度条显示当前的传送状态

# rc.local、rc.d、init.d Linux开机自动运行程序

* /etc/rc.local 该脚本是在系统初始化级别脚本运行之后再执行的，因此可以安全地在里面添加你想在系统启动之后执行的脚本

[root@test ~]# ll /etc/rc.local

lrwxrwxrwx. 1 root root 13 2月 28 2018 /etc/rc.local -> rc.d/rc.local #软链接到/etc/rc.d/rc.local

[root@test ~]# systemctl list-units --type=service |grep jenkins

jenkins.service loaded active running LSB: Jenkins Automation Server

* /etc/init.d init.d目录包含许多系统各种服务的启动和停止脚本。

[root@test ~]# ll /etc/init.d/

总用量 40

-rw-r--r--. 1 root root 13948 9月 16 2015 functions

-rwxr-xr-x 1 root root 5967 6月 8 2018 jenkins

-rwxr-xr-x. 1 root root 2989 9月 16 2015 netconsole

-rwxr-xr-x. 1 root root 6630 9月 16 2015 network

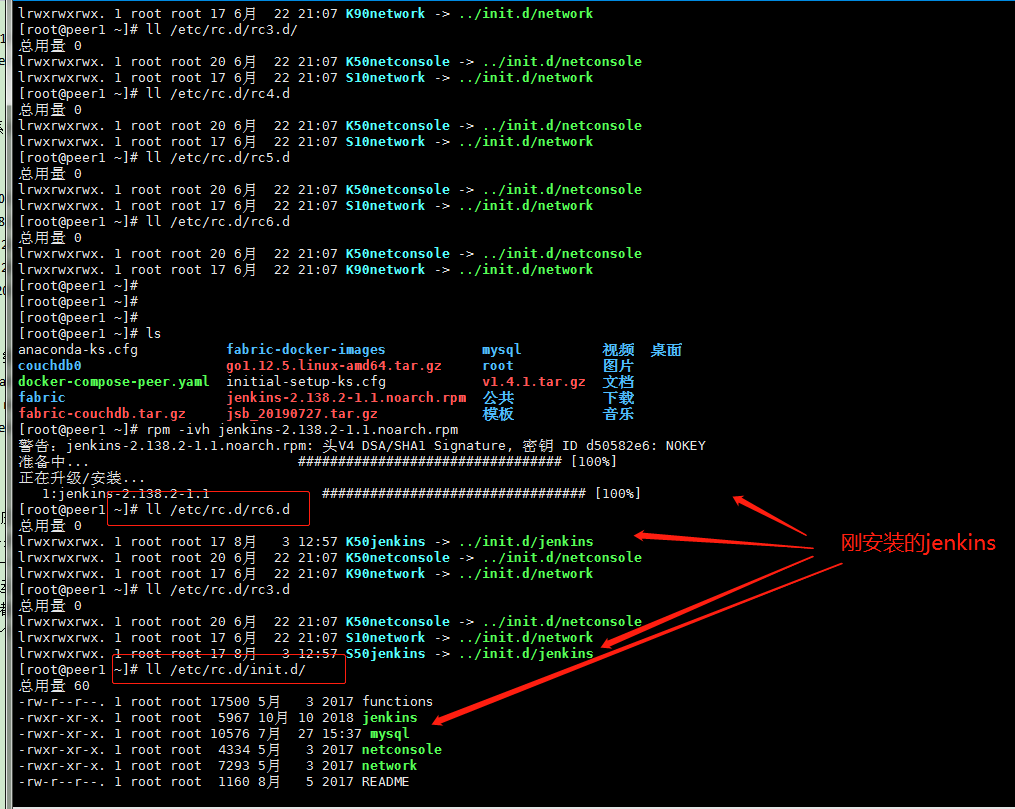
-rw-r--r--. 1 root root 1160 2月 1 2018 README

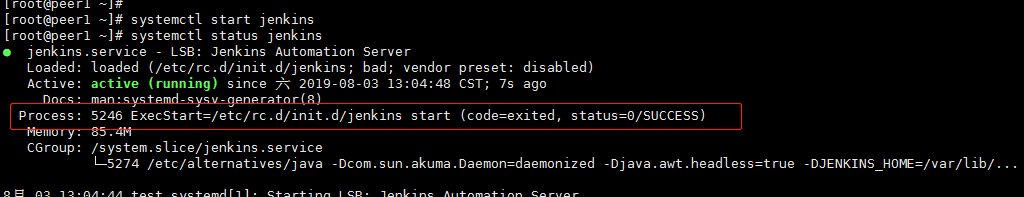
/etc/rc.d/init.d

该文件夹包含所有服务在各个运行等级中的全部启动脚本。一般来说，它们都是标准的shell脚本，遵守最基本的标准。每个脚本最少接受两个参数start和stop，它们分别代表启动和停止服务（如网页服务）。除此之外，init脚本通常还会接受一些额外的选项，如restart（重启服务器）、status（返回服务当前状态）、reload（告知服务从配置文件中重新载入配置）以及force-reload（强制服务重载它的配置）。当用不带参数的方式运行脚本的时候，一般应该返回一个它会接受的参数列表。

/etc/rc.d/rc0.d～/etc/rc.d/rc6.d

这些文件夹分别包含每个运行等级对应的init脚本。在实际使用中，它们一般通过符号链接到/etc/init.d文件夹下的实际文件。不过要注意的是，这些文件夹下的init脚本都有一些特别的名字，命名都以S（start）、K（kill）或D（disable）开头，后面跟一个数字。当init进入一个运行等级的时候，它会按照数字顺序运行所有以K开头的脚本并传入stop参数，除非对应的init脚本在前一个运行等级中没有启动。然后init按照数字顺序运行所有以S开头的脚本并传入start参数。任何以D开头的init脚本都会被忽略—这让你可以在指定的运行等级禁止一个脚本，或者你也可以仅仅移除全部符号链接。所以如果你有两个脚本，S01foo和S05bar，init首先会运行S01foo start，当它进入特定的运行等级后再执行S05bar start。





/etc/rc.d/rc.local

并非所有的发行版都使用了rc.local，通常它是一个留给用户修改的shell脚本。一般会在init进程结束的时候运行它，所以你可以在这里放一些想要运行的额外脚本，而不用再创建自己的init脚本。

/etc/rc.d/rc.sysinit

/etc/rc.d/rc.sysinit主要做在各个运行模式中相同的初始化工作，包括设定PATH、设定网络配置（/etc/sysconfig/network）、启动swap分区、设定/proc等等。

/etc/rc.d/rc

当运行级别改变时，负责启动/停止各种服务。

=====================================================================

注意：这只是启动脚本sh文件，只要脚本运行了，那我就可以在sh脚本里运行各种a.out程序！！！

各个运行级的详细解释：

0 为停机，机器关闭。 （千万别把initdefault设置为0，否则系统永远无法启动）

1 为单用户模式，就像Win9x下的安全模式类似。

2 为多用户模式，但是没有NFS支持。

3 为完整的多用户模式，是标准的运行级。

4 一般不用，特殊情况下可以用它来做一些事情。如在笔记本的电池用尽时，可以切换到这个模式来做一些设置。

5 就是X11，进到X Window系统了。

6 为重启，运行init 6机器就会重启。

0和6一般不用；

一、简单说明

　　/etc/init.d 是 /etc/rc.d/init.d 的软链接(soft link)。可以通过 ll 命令查看。

ls -ld /etc/init.d

lrwxrwxrwx. 1 root root 11 Aug 30 2015 /etc/init.d -> rc.d/init.d

　　都是用来放服务脚本的，当Linux启动时，会寻找这些目录中的服务脚本，并根据脚本的run level确定不同的启动级别。

　　在制作服务脚本的过程中，使用了Linux的两个版本，CentOS和Ubuntu，需要在两个版本中都可以开机启动服务。但Ubuntu没有 /etc/rc.d/init.d这个目录，所以，为了保持同一种服务在CentOS和Ubuntu使用的统一性，将服务脚本（注：服务脚本在两个不同版本中是不同的）都放在 /etc/init.d 目录下，最终达到的效果是相同的。

　　需要说明的是：在CentOS和Ubuntu两个版本中，除了服务脚本放置的目录是相同的，服务脚本的编写及服务配置都是不同的。比如CentOS使用Chkconfig进行配置，而Ubuntu使用sysv-rc-conf进行配置。

# 系统启动过程

　　1）BIOS自检 ，BIOS的功能由两部分组成，分别是POST码和Runtime服务。POST阶段完成后它将从存储器中被清除，而Runtime服务会被一直保留，用于目标操作系统的启动。BIOS两个阶段所做的详细工作如下：

　　步骤1：上电自检POST(Power-on self test)，主要负责检测系统外围关键设备（如：CPU、内存、显卡、I/O、键盘鼠标等）是否正常。例如，最常见的是内存松动的情况，BIOS自检阶段会报错，系统就无法启动起来；

　　步骤2：步骤1成功后，便会执行一段小程序用来枚举本地设备并对其初始化。这一步主要是根据我们在BIOS中设置的系统启动顺序来搜索用于启动系统的驱动器，如硬盘、光盘、U盘、软盘和网络等。我们以硬盘启动为例，BIOS此时去读取硬盘驱动器的第一个扇区(MBR，512字节)，然后执行里面的代码。实际上这里BIOS并不关心启动设备第一个扇区中是什么内容，它只是负责读取该扇区内容、并执行。

　　至此，BIOS的任务就完成了，此后将系统启动的控制权移交到MBR部分的代码。

　　2）系统引导，通常情况下，诸如lilo、grub这些常见的引导程序都直接安装在MBR中。详细过程，请自行google

　　3）启动内核，它首先会去解析grub的配置文件/boot/grub/grub.conf，然后加载内核镜像到内存中，并将控制权转交给内核。而内核会立即初始化系统中各设备并做相关的配置工作，其中包括CPU、I/O、存储设备等。

　　关于Linux的设备驱动程序的加载，有一部分驱动程序直接被编译进内核镜像中，另一部分驱动程序则是以模块的形式放在initrd(ramdisk)中。

　　 Linux内核需要适应多种不同的硬件架构，但是将所有的硬件驱动编入内核又是不实际的，而且内核也不可能每新出一种硬件结构，就将该硬件的设备驱动写入内核。实际上Linux的内核镜像仅是包含了基本的硬件驱动，在系统安装过程中会检测系统硬件信息，根据安装信息和系统硬件信息将一部分设备驱动写入 initrd 。这样在以后启动系统时，一部分设备驱动就放在initrd中来加载。这里有必要给大家再多介绍一下initrd这个东东：

　　 initrd 的英文含义是 bootloader initialized RAM disk，就是由 boot loader 初始化的内存盘。在 linu2.6内核启动前，boot loader 会将存储介质中的 initrd 文件加载到内存，内核启动时会在访问真正的根文件系统前先访问该内存中的 initrd 文件系统。在 boot loader 配置了 initrd 的情况下，内核启动被分成了两个阶段，第一阶段先执行 initrd 文件系统中的init，完成加载驱动模块等任务，第二阶段才会执行真正的根文件系统中的 /sbin/init 进程。

　　通过以上分析，grub的stage2将initrd加载到内存里，让后将其中的内容释放到内容中，内核便去执行initrd中的init脚本，这时内核将控制权交给了init文件处理。我们简单浏览一下init脚本的内容，发现它也主要是加载各种存储介质相关的设备驱动程序。当所需的驱动程序加载完后，会创建一个根设备，然后将根文件系统rootfs以只读的方式挂载。这一步结束后，释放未使用的内存，转换到真正的根文件系统上面去，同时运行/sbin/init程序，执行系统的1号进程。此后系统的控制权就全权交给/sbin/init进程了。

　　初始化系统，接下来就是初始化系统的工作了，/sbin/init进程是系统其他所有进程的父进程，当它接管了系统的控制权先之后，它首先会去读取/etc/inittab文件来执行相应的脚本进行系统初始化，如设置键盘、字体，装载模块，设置网络等。主要包括以下工作：

　　（1）执行系统初始化脚本(/etc/rc.d/rc.sysinit)，对系统进行基本的配置，以读写方式挂载根文件系统及其它文件系统，到此系统算是基本运行起来了，后面需要进行运行级别的确定及相应服务的启动。

　　（2）执行/etc/rc.d/rc脚本。该文件定义了服务启动的顺序是先K后S，而具体的每个运行级别的服务状态是放在/etc/rc.d/rc\*.d（\*=0~6）目录下，所有的文件均是指向/etc/init.d下相应文件的符号链接。rc.sysinit通过分析/etc/inittab文件来确定系统的启动级别，然后才去执行/etc/rc.d/rc\*.d下的文件。

/etc/init.d-> /etc/rc.d/init.d

/etc/rc ->/etc/rc.d/rc

/etc/rc\*.d ->/etc/rc.d/rc\*.d

/etc/rc.local-> /etc/rc.d/rc.local

/etc/rc.sysinit-> /etc/rc.d/rc.sysinit

　　我们以启动级别3为例来简要说明一下，/etc/rc.d/rc3.d目录，该目录下的内容全部都是以 S 或 K 开头的链接文件，都链接到"/etc/rc.d/init.d"目录下的各种shell脚本。S表示的是启动时需要start的服务内容，K表示关机时需要关闭的服务内容。/etc/rc.d/rc\*.d中的系统服务会在系统后台启动，如果要对某个运行级别中的服务进行更具体的定制，通过chkconfig命令来操作，或者通过setup、ntsys、system-config-services来进行定制。如果我们需要自己增加启动的内容，可以在init.d目录中增加相关的shell脚本，然后在rc\*.d目录中建立链接文件指向该shell脚本。这些shell脚本的启动或结束顺序是由S或K字母后面的数字决定，数字越小的脚本越先执行。例如，/etc/rc.d/rc3.d /S01sysstat就比/etc/rc.d/rc3.d /S99local先执行。

　　（3）执行用户自定义引导程序/etc/rc.d/rc.local。其实当执行/etc/rc.d/rc3.d/S99local时，它就是在执行/etc/rc.d/rc.local。S99local是指向rc.local的符号链接。就是一般来说，自定义的程序不需要执行上面所说的繁琐的建立shell增加链接文件的步骤，只需要将命令放在rc.local里面就可以了，这个shell脚本就是保留给用户自定义启动内容的。

　　（4）完成了系统所有的启动任务后，linux会启动终端或X-Window来等待用户登录。tty1,tty2,tty3...这表示在运行等级1，2，3，4的时候，都会执行"/sbin/mingetty"，而且执行了6个，所以linux会有6个纯文本终端，mingetty就是启动终端的命令。除了这6个之外还会执行"/etc/X11/prefdm-nodaemon"这个主要启动X-Window

至此，系统就启动完毕了。

接下来就是执行/bin/login程序，进入登录状态

二、 init.d目录包含许多系统各种服务的启动和停止脚本。

　　/etc/init.d里的shell脚本能够响应start，stop，restart，reload命令来管理某个具体的应用。比如经常看到的命令： /etc/init.d/networking start 这些脚本也可被其他trigger直接激活执行，这些trigger被软连接在/etc/rcN.d/中。这些原理似乎可以用来写daemon程序，让某些程序在开关机时运行。