**ipv4 176.121.204.40**

**路由 176.121.204.1**

**FTP 176.121.0.120 nsd1811 1811tedu.cn**

**• 命令行提示标识的含义**

**– [当前用户@主机名 当前所在位置]$**

**Day01**

**1.Linux 操作系统，用于服务器端**

**Unix 1970-1-1诞生**

**Linux内核 能调配所有的硬件 （管理CPU）**

**Linux系统为内核与各个外围软件的组合**

**2.**

**RHEL、CentOS**

1. **RHEL7**
2. **硬盘分区：**

**物理硬盘～分区规划～格式化～读写/文档**

**Windows文件系统：NTFS、FAT**

**Linux文件系统： ext4(RHEL6) xfs(RHEL 7)**

**swap(交换文件系统)[或称虚拟内存]：缓解内存的不足**

**Linux分区表示方法**

**/dev ——设备:hd,表示IDE设备 /dev/hd 如，/dev/hda1 /dev/hdb2..**

**sd，表示SCSI设备 /dev/sd**

**[当前用户@主机名 工作目录]# (root权限为#，普通用户为$)**

1. **/root 第一个/为根目录**
2. **Ls 中，蓝色为目录 黑色为文本文件**
3. **找文件时 两下tab可看到其目录下子文件或者文件夹**

**[root@room4pc27 ~]# cat /proc/meminfo --查看内存信息 memoryinformation**

**[root@localhost /]# cat -n /etc/passwd -- -n提供编号显示功能**

1. **ifconfig>>**

**lo：loopback 回环地址 127.0.0.1（永远代表本机）**

**此时，ping 127.0.0.1 或者127.1 都可；**

**virbr0：虚拟网络地址；**

**配置网络地址： ifconfig eth0 地址**

1. **相对路径和绝对路径**

**绝对路径：以根目录为开头 （cd /）**

**相对路径：以当前路径为参照，注意改变路径后的情况**

1. **head -k 文件**

**tail -k 文件**

**不设置k值时 二者默认为10行显示**

**---------------------------------- 命令格式：命令 选项 参数1 ..参数n**

**mount挂载**

**挂载操作：让目录成为设备的访问点**

**mount 设备路径 挂载点目录**

**mount /dev/cdrom /dvd**

**取消挂载（卸载） umount 挂载点目录**

**umount /dvd**

**----卸载光驱设备时提示“设备正忙”无法卸载,其原因是什么?**

**当前所在位置在挂载点内,有其他终端或用户正在访问**

**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**

**~表示用户的家目录 XX@localhost ~ ~即为XX**

1. **通配符**

**\* 任意多个字符**

**？ 单个字符**

**[] 匹配1或者多个**

**{} 全匹配**

1. **别名设置：**

**alias 别名名称=‘命令’**

**取消设置 ：**

**unalias 别名名称**

**别名优先级更高**

1. **ls -R xx**

**逐层递归展开**

**cp -r**

**mkdir -p**

**mv -r**

1. **mv 移动指令，也可以用以重命名**

**mv 文件 目标路径**

**若不改变路径，则可视为重命名**

**13.cp 命令中 -f权限反而不如-i 『特例』**

**若要强制覆盖，可以反斜杠 \cp ，意思是在本次命令中取消别名设置，以使**

**cp=‘cp -i’ 中， -i不生效（不产生提示）**

**保持源文件属性不变 cp -p**

**支持多个参数，永远把最后一个参数作为目标，其余作为源进行拷贝。**

**复制时可以重命名,同一目录复制**

1. **vim 可以创建文件 但不可创建不存在的目录。**
2. **设置永久别名**

**修改配置文件 /root/.bashrc 针对root用户**

**全局配置 /etc/bashrc 针对所有用户**

1. **下载软件包**

**wget URL**

**wget URL -O（字母o大写） 目标路径/文件名 ：相当于另存为**

**装包**

**rpm -q 软件名 查询软件包是否安装 后加软件名即可**

**rpm -ql 软件名 查询软件具体安装情况**

**rpm -ivh 软件完整包名字 repo 安装(加完整的包名.rpm)**

**rpm -e 软件名 卸载 后加软件名即可**

**rpm --import 软件 导入包**

**rpm -qpi 未安装的软件包信息**

**rpm不可自动解决软件依赖关系**

1. **yum软件仓库 自动解决软件依赖关系**

**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**

**众多的软件包 仓库数据文件 具备光盘所有内容**

**清单**

**客户端：虚拟机server**

**书写yum客户端配置文件，指定服务端位置**

**/etc/yum.repos.d/\*.repo**

1. **配置永久主机名 --> 修改配置文件 （创建hostname 直接写入）**

**配置永久IP地址/子网掩码/网关地址**

**-- 改配置文件 /etc/sysconfig/network-scripts/ifcfg-eth0**

**--使用命令行（三步）：查看网卡名 nmcli connection show**

**修改地址 nmcli connection modify ‘网卡名’**

**ipv4.method manual**

**ipv4.addresses ‘地址、网关’**

**connection.autoconnect yes (1行命令)**

**激活网卡 nmcli connection up ‘网卡名’**

**配置永久DNS地址(配置并且激活其他网络参数之后 最后配置DNS服务器地址)**

**/etc/resolv.conf**

**nslookup 测试DNS域名解析**

**网卡配置文件地址：**

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

1. **> 覆盖重定向**

**>> 追加重定向**

**&**

**20.**

**管道操作 | 顺序执行,前面的一个命令作为后一个命令的参数，输出最后一个结果**

**echo 1+1 | bc**

**cat -n /etc/passwd | head -12 | tail -5**

**head -12 /etc/passwd | tail -5 | cat -n**

**NTP:Network Time Protocol 网络时间协议**

**chrony**

**配置文件：/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**

**服务名：chronyd**

**daemon:守护神;（希腊神话中）半人半神的精灵;[计]守护进程**

1. **组的设置方便对于用户的管理 对于权限的管理 GID**

**一个用户必须至少属于一个组**

**所以每创建一个用户，Linux便会创建一个同名组**

**root下，临时登陆用户 ：su - 用户名**

**usermod -u UID -d dir -s shells -G group:修改用户信息**

1. **/etc/passwd 存放用户信息**

**各部分代表的信息 root:x:0:0:root:/root:/bin/bash**

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

**/etc/shadow 存放密码信息**

**用户名：密码加密字符串 ：上一次修改密码的时间：**

**上一次修改密码的时间-->1970-1-1到目前时间经过的天数。**

**gpasswd -a -d 用户名 组名 :管理组：添加/删除用户**

**id 用户名 进行验证**

**四个文件：**

**/etc/passwd**

**/etc/shadow**

**/etc/group ：（具体表示） 组名:组的密码占位符:组的GID:组的成员列表**

**/etc/gshadow**

**24.tar解压归档文件**

**tar 选项 压缩包的名字 被压缩的原文档**

**tar -zcf 对应后缀 .gz**

**tar -jcf .bz2**

**tar -Jcf .xz**

**tar -xf 压缩包 -C 目标目录 ：释放到指定目录**

**tar -tf 压缩包 ： 查看文档清单**

**zip归档 ：**

**zip -r /路径/压缩包 /路径/被压缩的文件**

**unzip /路径/压缩包 -d 释放路径**

**25.date -s ‘’：修改系统时间，可进行NTP服务器设置时间同步**

**vim /etc/chrony.conf**

**systemctl restart chronyd**

**systemctl enable chronyd**

1. **基本权限**

**r w x 4 2 1**

**对于文本文件**

**r:cat less tail head**

**w:vim > >>**

**x:shell脚本 、cd切换进该目录 chmod +x**

**3个权限排列顺序：rwx**

**开头：- 文本文件 d 目录 l快捷方式**

**权限：所有者>所属组>其他人 u> g >o**

**ls -ld 按顺序排列，每个群体三个权限符号 后面接 属主A（所有者）、所属组B，某一用户不是A或不属于组B，则身份为 其他人o ！**

**drwxrwxrwx. 2 root root 6 12月 6 10:14 /nsd01/**

**w中，mv cp等可以修改目录内容，但是无法改变目录本身，如重命名**

**一般 rx 同步出现**

**chmod u+-=，g+-=，o+-= /etc 改权限**

**chown [-R] 所有者：所属组 源文件/目录 修改归属关系**

**（单独修改所属组时需要加：）**

**注意：**

**管理员root具备一切权限**

**Linux判别权限的方法： 匹配及停止原则**

**1. 查看用户对于文档的身份，属于那个归属关系 所有者>所属组>其他人**

**2. 查看相应归属关系位置的权限**

**-------------》》判断能否访问**

**附加权限1 Set GID**

**位置 覆盖所属组的x位上，原有x权限则显示s 无x权限则显示S**

**主要作用：传递父目录的所属组身份。 并且是作用于新增的目录或者文件，不影响设置Set GID之前，已存在的。**

**附加权限2 Set UID**

**传递所有者身份**

**所有者的x位上，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**

**附加权限3 Sticky Bit**

**其他人o的执行权限x位上,其他人的权限标识会变为 t**

**– 适用于开放 w 权限的目录,可以阻止用户滥用 w 写入**

**权限(禁止操作别人的文档)**

**acl权限**

**setfacl -m u:haha:rx /opt/test 增加权限，立即生效**

**getfacl /opt/test 查看权限**

**setfacl -x setfacl -x u:dc /nsd11 删除指定的ACL**

**setfacl -b setfacl -b /nsd11 清除所有ACL**

1. **LDAP:用户集中管理**

**网络用户，由网络中的LDAP服务器统一提供用户信息**

**使用LDAP认证**

**什么是LDAP?**

**• 轻量级目录访问协议**

**– Lightweight Directory Access Protocol**

**– 由服务器来集中存储并向客户端提供的信息,存储方**

**式类似于DNS分层结构**

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

**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.安装软件sssd,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**

**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**

**NFS:Network File System,网络文件系统**

**查看NFS资源**

**– showmount -e [服务器地址]**

**nfs挂载 ：mount classroom.example.com:/home/guests/ /home/guests**

**配置文件 /etc/exports**

**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)**

**/content/ 172.25.254.0/24(ro)**

**28.**

**cron计划任务**

**crontab 任务记录的编写 ：**

**分 时 日 月 周 任务命令行**

**\* :匹配范围内任意时间**

**, :分隔多个不连续的时间点**

**- :指定连续时间范围**

**/n :指定时间频率,每n ...**

**如 \*/2 每两个（）执行一次**

**• 使用 crontab 命令**

**– 编辑:crontab -e [-u 用户名]**

**– 查看:crontab -l [-u 用户名]**

**– 清除:crontab -r [-u 用户名]**

**任务文件 /var/spool/cron/用户名**

1. **grep拓展 (查找文本内容)**

**显示文件的内容，有效的配置（去除注释以#开头的行、去除空行）**

**# grep -v ^# /etc/default/useradd**

**# grep -v ^# /etc/default/useradd | grep -v ^$**

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

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

**grep -i 忽略大小写**

**-v 取反查找**

**^a 以a开头 以a结尾 a$**

1. **查找文件**

**根据预设的条件递归查找对应的文件**

**find**

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

**– 常用条件表示:**

**-type 类型(f文件、d目录、l快捷方式)**

**-name "文档名称"**

**-size +|-文件大小(k、M、G)**

**-user 用户名**

**31.**

**查找文件**

**• 使用find命令的 -exec 操作**

**– find .. .. -exec 处理命令 {} \;**

**– 优势:以 {} 代替每一个结果,逐个处理,遇 \; 结束**

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

**# find /root/ -name "nsd\*" -type f -exec cp {} /opt \;**

**# ls /opt/**

**案例1:为虚拟机 server 配置以下静态地址参数**

**– 主机名:server0.example.com**

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

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

**server0.example.com**

**– IP地址:172.25.0.120**

**– 子网掩码:255.255.255.0**

**– 默认网关:172.25.0.254**

**[root@server0 ~]# nmcli connection modify 'System eth0' ipv4.method manual**

**ipv4.addresses '172.25.0.120/24 172.25.0.254' connection.autoconnect yes**

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

**– DNS服务器:172.25.254.254**

**# echo nameserver 172.25.254.254 > /etc/resolv.conf**

**# cat /etc/resolv.conf**

**常见命令**

**[root@A /]# cat /etc/redhat-release #显示系统版本**

**• 列出CPU处理器信息**

**[root@room9pc01 ~]# lscpu**

**• 检查内存大小、空闲情况**

**[root@room9pc01 ~]# cat /proc/meminfo**

**[root@A /]# uname -r 列出内核版本**

**127.0.0.1:永远代表本机**

**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**

**mount挂载操作：让目录成为设备的访问点**

**光驱设备 /dev/cdrom**

**– 格式:mount 设备路径 挂载点目录**

**mount /dev/cdrom /dvd**

**卸载：当前路径不要是挂载点目录**

**umount /dvd 卸载**

**ls -l -h -A -d**

**mkdir -p 递归地创建目录**

**mv 移动/改名**

**• mv — Move**

**– 格式:mv [选项]... 原文件... 目标路径**

**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**

**cp -r /boot/ /opt/ #复制目录必须要加上-r选项**

**临时取消别名（-i询问） [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**

**真机执行ssh进行远程管理虚拟机**

**格式： ssh -X root@对方的IP地址**

**远程传输文件 scp 源文件 root@对方的IP地址:/root/ (保存路径)**

**wget• 使用wget下载工具**

**– wget 软件包的URL网址 #默认下载到当前路径下**

**-O(大写)：下载另存为**

**• RPM Package Manager,RPM包管理器**

**– rpm -q 软件名... #查询软件包是否安装**

**– rpm -ivh 软件名-版本信息.rpm... #安装软件包**

**– rpm -e 软件名... #卸载程序**

**rpm -ql vsftpd #查询已安装软件的安装清单**

**配置永久的IP地址、子网掩码、网关地址**

**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**

**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'**

**永久配置DNS服务器地址(在配置网络参数时，最后配置DNS服务器地址)**

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

**nameserver 172.25.254.254**

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

**测试DNS解析命令**

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

**[root@server0 ~]# nslookup desktop0.example.com**

**date -s ‘’修改系统时间**

**管道操作 |：**

**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**

**ifconfig | head -2 >1.txt**

**添加用户**

**• 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 -g 所属组**

**-s /sbin/nologin:禁止用户登陆系统**

**用户信息文件 /etc/passwd 具体含义：**

**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**

**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**

**删除用户**

**• 使用 userdel 命令**

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

**添加组**

**• 使用 groupadd 命令**

**– groupadd [-g 组ID] 组名**

**组文件：/etc/group 组名:组的密码占位符:组的GID:组的成员列表\**

**管理组成员**

**组密码信息存放在 /etc/gshadow 文件**

**• 使用 gpasswd 命令**

**– gpasswd -a 用户名 组名**

**– gpasswd -d 用户名 组名**

**修改组属性**

**• 使用 groupmod 命令**

**– groupmod [-g 组ID] [-n 新组名] 组名**

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

**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命令：**

**tar 选项 压缩包的名字 被压缩的源文档 ......**

**tar -zcf 压缩包的名字 被压缩的源文档 ......**

**tar -jcf 压缩包的名字 被压缩的源文档 ......**

**tar -Jcf 压缩包的名字 被压缩的源文档 ......**

**zcf对应 .gz**

**jcf对应 .bz2**

**Jcf对应 .xz**

**解压：tar -xf 压缩包的名字 -C 释放的路径**

**查看tar包数据内容**

**tar -tf tar包名**

**NTP网络时间同步，配置客户机：**

**配置文件**

**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**

**之后记住重启程序(服务)！**

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

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

**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**

**chown 用户名：组名 源文件 设置源文件的归属**

**1.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**

**2.如何格式化 swap 交换文件系统,如何启用 swap 分区,如何查看 swap 空间成员信息**

**mkswap**

**swapon**

**swapon -s**

**默认的解释器：/bin/bash**

**主机名 /etc/hostname**

**DNS服务器配置文件 /etc/reslov.conf**

**别名配置文件 /root/.bashrc**

**yum客户端配置文件 /etc/yum.repos.d/**

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

1. **一个扇区默认大小 512字节**

**(MBR 主引导记录模式 main boot record:最大支持空间2.2Tb**

**分区类型：主分区 扩展分区 逻辑分区**

**一般三个主分区 ，但最多可以4个主分区**

**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)**

**识别硬盘 分区规划 格式化 挂载使用**

**2. lsblk 查看系统磁盘设备**

**fdisk /dev/.. 进行分区**

**刷新分区情况 partprobe**

**mkfs.xfs 格式化生成文件系统 blkid 查看文件系统**

**df -h #查看正在挂载的分区使用情况**

**挂载使用，手工挂载**

**或者设置开机自动挂载设备**

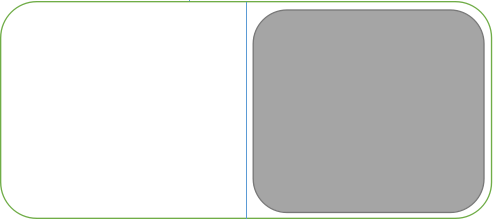
**配置文件 /etc/fstab**

**• 修改/etc/fstab配置文件**

**file system table**

**• 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**

****

**LVM逻辑卷**

**作用： 1.整合分散的空间**

**2.逻辑卷空间可以扩大**

**新建逻辑卷： 将众多的物理卷(pv)组成卷组(vg)，再从卷组中划分逻辑卷(lv)**

**创建逻辑卷pv-->vg -->lv**

**删除逻辑卷lv-->vg-->pv（可以不删除pv）**

**part1.**

**创建卷组**

**命令格式: vgcreate 卷组名 设备路径......**

**part2.**

**创建逻辑卷**

**命令格式:lvcreate -L 大小 -n 逻辑卷名称 基于的卷组**

**lvs #查看逻辑卷信息**

**vgs #查看卷组信息**

**先扩展逻辑卷空间：**

**lvextend -L 10G /dev/...**

**再扩展文件系统：**

**ext4：resize2fs /dev...**

**xfs：xfs\_growfs**

**(如果卷组vg空间不够 要先扩展卷组 vgextend)**

**ext4文件系统支持逻辑卷空间缩小，xfs文件系统不支持。**

**PE：卷组划分空间的单位**

**• 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 空闲分区...**

**[root@server0 ~]# vgdisplay nsd #显示卷组详细信息,看PE的大小**

**vgcreate -s 1M nsd 设备路径 #创建卷组的时候直接设置PE为1M**

**[root@server0 ~]# vgchange -s 1M nsd #修改卷组PE的大小**

**[root@server0 ~]# vgdisplay nsd #显示卷组详细信息,看PE的大小**

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

**[root@server0 ~]# lvcreate -l 50 -n lvtest03 nsd**

**[root@server0 ~]# lvs**

**-l:指定PE的个数**

1. **/root/.bashrc**

**/etc/bashrc**

1. **单引号：取消特殊字符的含义，只输出字符串**

**反撇号 `` 或$():将命令的输出**

**双引号 “” 可以嵌套可使用的命令等**

1. **环境变量 USER：当前登录的用户名**

**执行脚本：A.加x权限，（一般绝对）路径执行**

**B.sh (-x) 脚本文件路径**

**C. source 脚本文件路径**

**D. . 脚本文件路径**

**B C D 不需要权限，可以用相对路径，D记住.后面加空格**

**位置变量 由系统定义赋值完成，用户直接调用**

**方便向脚本中传递命令行参数(按需要跟在脚本执行命令后面输入)**

**> :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**

**将 echo hello 的输出信息,变成错误输出**

**echo hello > &2**

**exit 2 #改变返回值**

**预定义变量**

**$0 当前所在的进程或者脚本名**

**$$ 当前运行的进程的PID**

**$# 已加载的位置变量的个数**

**$\* 所有位置变量的值**

**$? 程序退出后的状态值,0表示正常,其他值异常**

**$! Shell最后运行的后台Process的PID**

**[root@server0 ~]# cat 1.sh**

**#!/bin/bash**

**echo $1**

**echo $2**

**echo $3**

**echo 你一共输入了$#个位置变量**

**echo 他们分别是$\***

**排序：**

**#!/bin/bash**

**read -p '请输入第一个整数：' num1**

**read -p '请输入第二个整数：' num2**

**read -p '请输入第三个整数：' num3**

**if [ $num1 -gt $num2 ];then**

**a=$num2**

**num2=$num1**

**num1=$a**

**fi**

**if [ $num2 -gt $num3 ];then**

**b=$num3**

**num3=$num2**

**num2=$b**

**fi**

**if [ $num1 -gt $num2 ];then**

**c=$num2**

**num2=$num1**

**num1=$c**

**fi**

**echo $num1 $num2 $num3**

**（vim下，dd 删除当前行 yy 复制当前行**

**d$或者C 删除至光标前）**

**常用的测试选项**

**• 检查文件状态**

**-e:文档存在为真**

**比如[ -e /etc ] echo $? 0**

**-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**

**[ -f /etc/etc ] echo $? 1**

**-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**

**Selinux**

**setenforce 0 临时设为permissive宽松**

**setenforce 1 临时设为enforcing 强制**

**查看当前模式:getenforce**

**固定配置:/etc/selinux/config 文件**

**防火墙：**

**匹配规则的原则： 匹配及停止**

**预设安全区域 四个**

**• 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，不给回应，节省资源**

**所以drop状态下ping会超时（得不到回应）**

**数据包 ：源IP地址 目标IP地址 数据 目标端口号**

**防火墙判定的规则:**

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

**2.进入默认区域public**

**端口,是一种标识 root可以改变端口号，一个程序可以具备多个端口**

**一些服务具有默认端口**

**默认端口号**

**http:80**

**https:443**

**ftp:21**

**tftp:69**

**DNS:53**

**telnet:23**

**SMTP:25**

**pop3:110**

**SNMP:161**

**防火墙配置**

**具体一些相关命令：**

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

**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**

**在public区域添加服务或协议**

**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**

**单独拒绝某一主机（IP）**

**firewall-cmd --zone=block --add-source=172.25.0.10**

**firewall-cmd --reload**

**firewall-cmd --zone=block --list-all**

**firewall-cmd --zone=block --remove-source=172.25.0.10 #删除该规则**

**firewall-cmd --permanent --zone=public --add-service=http #永久添加**

**协议**

**firewall-cmd --permanent --zone=public**

**--add-forward-port=port=5423:proto=tcp:toport=80 #重定向端口访问**

**firewall-cmd --reload #重新加载防火墙所有规则**

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

**ftp文件传输协议**

**装包 vsftpd**

**设备路径 挂载点 文件系统 参数 备份标记 检测顺序**

**//172.25.0.11/common (common为共享名) /mnt/nsd cifs defaults,user=harry,pass=123,\_netdev 0 0**

**samba：**

**服务端:**

**1.设置防火墙默认区域为trusted**

**2.安装软件包samba**

**3.创建共享用户的帐号 pdbedit -a**

**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**

**掛載點爲共享名 「共享名」**

**6.重起smb服务**

**7.开放SELinux读写布尔值 setsebool samba\_expore\_all\_rw on**

**8.设置本地目录的权限 setfacl**

**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**

**//172.25.0.11/devops /mnt/dev cifs**

**defaults,user=chihiro,pass=123,\_netdev 0 0**

**nfs共享2： Linux与Linux平台之间的共享**

**A.服务端修改配置文件 /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)**

**/content/ 172.25.254.0/24(ro)**

**重启服务 nfs-server**

**B.客户端自动挂载**

**/etc/fstab**

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

**mount -a**

**df -hT**

**ipv6 128个二进制位,冒号分隔 的十六进制数**

**ISCSI网络磁盘（默认端口号3260）**

**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target磁盘组 （命名规范）**

**c:进行关联lun /tpg1/luns**

**d:配置ACL访问控制,客户端声称的名字 /tpg1/acls, acls create AA**

**e:启用网络接口,启用端口 /tpg1/portals**

**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 /etc/iscsi/initiatorname.iscsi**

**名字与服务端d中acls名字保持一致！ 例如 改为AA**

**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(man iscsiadm 搜索example改ip)**

**6.重起客户端服务iscsi,识别加载共享存储**

**iqn.yyyy-mm.倒序域名:自定义标识**

**命名作用：用来识别 target 磁盘组,也用来识别客户机身份**

**mariadb 默认端口 3306**

**mysqladmin [-u用户名] [-p[旧密码]] password '新密码' 设置密码**

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

**针对于表**

**select from**

**insert into 表 values（字段）**

**update 「數據庫」.表名 set 字段=新值 where**

**delete from 「數據庫」.表名 where**

**授权管理：**

**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';**

**web虚拟机：**

**DocumentRoot:设置网页文件存放的根目录(/var/www/html)**

**未来应该学会：**

**LAMP: Linux Apache MySQL/MariaDB PHP/Python**

**LNMP: Linux Nginx MySQL/MariaDB PHP/Python**

**httpd的主配置文件 /etc/httpd/conf/httpd.conf**

**网络路径与服务器实际存放路径:网页文件的根目录**

**示例：**

**A.网络路径**

**# firefox 172.25.0.11-->httpd-->DocumenRoot-->/var/www/myweb**

**服务器实际存放路径**

**/var/www/myweb**

**B.网络路径**

**# firefox 172.25.0.11/var/www/myweb/nsd---->httpd**

**服务器实际存放路径**

**/var/www/myweb/var/www/myweb/nsd**

**C.网络路径**

**# firefox 172.25.0.11/nsd**

**服务器实际存放路径**

**/var/www/myweb/nsd**

**虚拟Web主机**

**– 由同一台服务器提供多个不同的Web站点**

**三种构建方式（区分方式）**

**– 基于域名的虚拟主机**

**– 基于端口的虚拟主机**

**– 基于IP地址的虚拟主机**

**• 配置文件路径**

**– /etc/httpd/conf/httpd.conf #主配置文件**

**– /etc/httpd/conf.d/\*.conf #调用配置文件**

**两点：一旦使用了虚拟Web主机的功能,所有的站点都必须,要使用虚拟web主机功能呈现，即主配置文件中的书写可能失效；**

**虚拟Web主机,会由上及下进行匹配,写在上面的优先级高。**

**书写格式（调用配置文件）：**

**<VirtualHost \*:80> #在所有IP上80端口,启用虚拟Web主机**

**ServerName www0.example.com #指定网站访问域名**

**DocumentRoot /var/www/baidu #指定存放网页文件路径**

**</VirtualHost>**

**客户机地址限制**

**• 使用 <Directory> 配置区段**

**– 每个文件夹自动继承其父目录的ACL访问权限**

**– 除非针对子目录有明确设置**

**主配置文件默认禁止访问除了/var/www 下的所有目录（有单独设置权限）**

**因此改变成新的网页文件根目录时需要针对进行访问配置。**

**动态网站的部署：由动态程序转后端模块处理，服务端的原始网页 ≠ 浏览器访问到的网页。**

1. **页面跳转 (页面别名 地址重写)**

**配置: Alias 网络路径 实际服务器本地路径**

**当客户端访问 网络路径 ,页面跳转到 实际服务器本地路径**

**当客户端访问 浏览器输入的路径为/nsd ,页面跳转到/var/www/html/abc.html**

**Alias / /var/www/qq/webinfo.wsgi**

**B.安装软件mod\_wsgi,翻译Python页面的代码**

**vim /etc/httpd/conf.d/nsd01.conf**

**<VirtualHost \*:80>**

**ServerName webapp0.example.com**

**DocumentRoot /var/www/qq**

**WsgiScriptAlias / /var/www/qq/webinfo.wsgi**

**</VirtualHost>**

**C.此虚拟主机侦听在端口8909**

**[root@server0 /]# vim /etc/httpd/conf.d/nsd01.conf**

**Listen 8909 #让httpd程序监听8909**

**<VirtualHost \*:8909> #该虚拟Web主机利用8909提供**

**ServerName webapp0.example.com**

**DocumentRoot /var/www/qq**

**WsgiScriptAlias / /var/www/qq/webinfo.wsgi**

**</VirtualHost>**

**生效需要修改Selinux的策略, 非默认端口开放**

**semanage port -l | grep http**

**]# semanage port -a -t http\_port\_t -p tcp 8909**

**-a:添加 -t:类型 -p:协议**

**]# systemctl restart httpd**

**]# firefox webapp0.example.com:8909**

**（在末行模式 :set nu #开启行号功能）**

**基础邮件服务**

**SMTP:用户发邮件协议 25**

**pop3:用户收邮件协议 110**

**软件包 postfix pop3 110**

**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**

**parted分区工具：**

**GPT模式**

**命令如下：**

**parted /dev/vdb**

**（parted) mktable gpt #指定分区模式**

**(parted) print #输出分区表**

**(parted) mkpart #划分新的分区**

**交换空间**

**什么是交换空间**

**• 相当于虚拟内存**

**– 当物理内存不够用时,使用磁盘空间来模拟内存**

**– 在一定程度上缓解内存不足的问题**

**– 交换分区:以空闲分区充当的交换空间**

**格式化交换文件系统 mkswap /dev/vdb1**

**blkid /dev/vdb1**

**启用交换分区 swapon /dev/vdb1**

**swapon -s #查看组成交换空间成员信息**

**swapoff /dev/vdb1 #停用交换分区**

**开机自动启用交换分区**

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

**/dev/vdb1 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**

**不需要挂载点,开机自动挂载时挂载点直接写为swap**

**配置聚合连接：**

**热备份(activebackup)连接冗余**

**1.创建虚拟网卡team0 参考: man teamd.conf #全文搜索/example**

**]# nmcli connection add type team**

**ifname team0 con-name team0 autoconnect yes**

**config '{"runner": {"name": "activebackup"}}'**

**nmcli connection 添加 类型 team(组队)**

**网卡名 team0 配置文件名 team0 每次开机自动启用**

**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**

**生成的网卡配置文件信息**

**2. 添加成员**

**]# 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 添加 类型为 team的成员**

**配置文件名 team0-1 网卡为 eth1 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**

**删除指令 :**

**nmcli connection delete team0**

**nmcli connection delete team0-1**

**nmcli connection delete team0-2**

**######################################################**

**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**

**Service：**

1. **搭建主机，配置网络yum：**

**①默认情况，教学环境。真机可以直接与clone-vm7创建的虚拟机通信，虚拟机eth0的ip地址必须为192.168.4.0/24网络中的ip地址。**

**②设置yum仓库：服务端：真机搭建Web服务，共享光盘所有内容**

**mount /iso/rhel-server-7.4-x86\_64-dvd.iso /var/www/html/rhel7/**

**（可以写入 /etc/fstab）文件系统 iso9660**

**/var/lib/libvirt/images/iso/rhel-server-7.4-x86\_64-dvd.iso /var/www/html/rhel7 iso9660 defaults 0 0**

**③客户端： 虚拟机A 虚拟机B**

**[root@svr7 ~]# vim /etc/yum.repos.d/rhel7.repo**

**[rhel7]**

**name=rhel7**

**baseurl=http://192.168.4.254/rhel7**

**enabled=1**

**gpgcheck=0**

**上述也可使用ftp服务 （vsftpd）**

1. **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 存放内存中的映射数据,不占用磁盘**

**/tmp 存放系统运行过程中使用的一些临时文件**

**3.• 权限的数值化**

**– 基本权限:r = 4,w = 2,x = 1**

**– 附加权限:SUID = 4,SGID = 2,Sticky Bit = 1**

**4.du -sh**

**• du,统计文件的占用空间**

**– du [选项]... [目录或文件]...**

**– -s:只统计每个参数所占用的总空间大小**

**– -h:提供易读容量单位(K、M等)**

**5.• date,查看/调整系统日期时间**

**– date +%F 输出年月日**

**date +%R 输出当前时间**

**– date +"%Y-%m-%d %H:%M:%S" 输出完整格式的年月日加时间**

**– date -s "yyyy-mm-dd HH:MM:SS" 修改时间**

**6.• ln,创建软连接(快捷方式)**

**– ln -s 原始文件或目录 软连接文件**

**若原始文件或目录被删除,连接文件将失效**

**软连接可存放在不同分区/文件系统,支持目录**

**• ln,创建硬链接**

**– ln 原始文件 硬连接文件**

**若原始文件被删除,连接文件仍可用**

**硬链接与原始文件必须在同一分区/文件系统,不支持目录**

1. **zip**

**归档工具(跨平台)**

**• 归档+压缩操作**

**– zip [-r] 备份文件.zip 被归档的文档...**

**• 释放归档+解压操作**

**– unzip 备份文件.zip [-d 目标文件夹]**

1. **发布网络yum源**

**创建仓库数据文件(描述仓库所有包的数量及信息) createrepo /tools/other**

**书写客户端配置文件**

**客户端是读取的Yum仓库数据文件**

**9.开关参数的控制**

**:set nu|nonu 显示/不显示行号**

**:set ai|noai 启用/关闭自动缩进**

**字符串替换**

**:s/old/new 替换光标所在,当前行第一个“old”**

**:s/old/new/g 替换光标所在,当前行所有的“old”**

**:1,20s/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**

**vim不能创建路径**

1. **DNS解析**

**所有的域名必须以点结尾 [www.baidu.com.](http://www.baidu.com.) 一级域名在后面**

**完整合格的主机名FQDN: 主机头部+域名 站点名.域名后缀**

**yum -y install bind-chroot bind ：伯克利 Internet 域名服务，bind-chroot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**

**☆配置DNS服务器：**

**服务端：yum -y install bind-chroot bind**

**先备份主配置文件，而后进行修改 （养成好的习惯）**

**cp /etc/named.conf /etc/named.bak**

**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**

**}; 注意分号和大括号！**

**在上述地址库存放路径下建立地址库文件**

**]# cd /var/named**

**]# cp -p named.localhost tedu.cn.zone #保持权限进行复制**

**]# 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 #指定本区域DNS服务器为svr7，**

**svr7 A 192.168.4.7 然后地址为192.168.4.7（svr7.tedu.cn）**

**www A 192.168.4.1 #把www.tedu.cn正向解析为192.168.4.1 ftp A 192.168.4.2**

**多区域的DNS服务器构建：**

**1.修改主配置文件**

**[root@svr7 ~]# vim /etc/named.conf**

**options {**

**directory "/var/named";**

**};**

**zone "tedu.cn" IN {**

**type master;**

**file "tedu.cn.zone";**

**};**

**zone "qq.com" IN {**

**type master;**

**file "qq.com.zone";**

**};**

**3.建立地址库文件/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 #保持权限进行复制**

**]# vim /var/named/qq.com.zone**

**qq.com. NS svr7**

**svr7 A 192.168.4.7**

**www A 1.2.3.4**

**4.重起named服务**

**[root@svr7 /]# systemctl restart named**

**常见的DNS资源解析记录有哪些?**

**NS（指定本区域DNS服务器） A(正向解析，还有反向解析) CNAME(别名)**

**特殊的解析记录：**

1. **基于DNS的轮询**
2. **泛域名解析 \***

**[root@svr7 /]# vim /var/named/tedu.cn.zone**

**tedu.cn. NS svr7**

**svr7 A 192.168.4.7**

**tedu.cn. A 10.20.30.40**

**\* A 11.12.13.14**

**[root@svr7 /]# systemctl restart named**

**虚拟机B测试:**

**[root@pc207 /]# nslookup tedu.cn**

**[root@pc207 /]# nslookup wwww.tedu.cn**

**[root@pc207 /]# nslookup xixi.tedu.cn**

**有规律的泛域名解析**

**pc1.tedu.cn ----> 192.168.10.1**

**pc2.tedu.cn ----> 192.168.10.2**

**pc3.tedu.cn ----> 192.168.10.3**

**........**

**pc50.tedu.cn ----> 192.168.10.50**

**内置函数: $GENERATE 起始点-结束点**

**[root@svr7 /]# vim /var/named/tedu.cn.zone**

**$GENERATE 1-50 pc$ A 192.168.10.$**

**4.解析记录的别名**

**tts CNAME ftp**

**tts解析结果与ftp一致**

**[root@pc207 ~]# nslookup tts.tedu.cn**

**Server: 192.168.4.7**

**Address: 192.168.4.7#53**

**tts.tedu.cn canonical name = ftp.tedu.cn.**

**Name: ftp.tedu.cn**

**Address: 192.168.4.2**

**DNS子域授权：让父域的DNS服务器,可以解析子域DNS负责的域名**

**1.首先要在子域即客户端pc207装包bind bind-chroot**

**2.修改主配置文件**

**[root@pc207 ~]# cp /etc/named.conf /etc/named.bak**

**[root@pc207 ~]# vim /etc/named.conf**

**options {**

**directory "/var/named";**

**};**

**zone "bj.qq.com" IN {**

**type master;**

**file "bj.qq.com.zone";**

**};**

**3.建立地址库文件/var/named/bj.qq.com.zone**

**]# vim /var/named/bj.qq.com.zone**

**bj.qq.com. NS pc207**

**pc207 A 192.168.4.207**

**www A 192.168.100.1**

**4.重起named服务**

**[root@svr7 /]# systemctl restart named**

**5.测试解析**

**[root@pc207 /]# nslookup www.bj.qq.com 192.168.4.207**

**转到服务端**

**虚拟机A:修改地址库文件**

**[root@svr7 /]# vim /var/named/qq.com.zone**

**qq.com. NS svr7**

**bj.qq.com. NS pc207**

**svr7 A 192.168.4.7**

**pc207 A 192.168.4.207**

**www A 1.2.3.4**

**[root@svr7 /]# systemctl restart named**

**[root@svr7 /]# nslookup www.bj.qq.com 192.168.4.7**

**Server: 192.168.4.7**

**Address: 192.168.4.7#53**

**Non-authoritative answer: #非权威解答**

**Name: www.bj.qq.com**

**Address: 192.168.100.1**

**递归解析:**

**将首选DNS服务器,与其他DNS服务器交互,最终将解析结果,带回来的过程**

**迭代解析:**

**将首选DNS服务器,与其他DNS服务器交互**

**[root@svr7 /]# vim /etc/named.conf**

**options {**

**directory "/var/named";**

**recursion no; #禁止递归解析**

**};**

**一些细节点：**

**A.主机名映射文件/etc/hosts**

**作用:只为本机直接提供解析结果**

**[root@svr7 /]# vim /etc/hosts**

**192.168.4.222 www.sina.com**

**[root@svr7 /]# ping -c 1 www.sina.com**

**客户端解析域名优先级(匹配及停止)**

**1./etc/hosts**

**2./etc/resolv.conf**

**B.缓存DNS服务器:缓存解析记录,加快解析**

**在真机上操作:**

**1.真机搭建Yum仓库**

**]# cat /etc/redhat-release**

**]# mkdir /dvd**

**]# mount /iso/CentOS-7-x86\_64-DVD-1708.iso /dvd/**

**]# ls /dvd**

**]# cd /etc/yum.repos.d/**

**]# mkdir repo**

**]# mv \*.repo repo**

**]# vim dvd.repo**

**[dvd]**

**name=CentOS7.4**

**baseurl=file:///dvd**

**enabled=1**

**gpgcheck=0**

**]# yum -y install bind bind-chroot**

**2.查看真机DNS服务器**

**[root@room9pc01 /]# cat /etc/resolv.conf**

**# Generated by NetworkManager**

**search tedu.cn**

**nameserver 172.40.1.10**

**[root@room9pc01 /]#**

**3.真机修改配置文件**

**[root@room9pc01 /]# cp /etc/named.conf /etc/named.bak**

**[root@room9pc01 /]# vim /etc/named.conf**

**options {**

**directory "/var/named";**

**forwarders { 172.40.1.10; }; #转发给172.40.1.10**

**};**

**[root@room9pc01 /]# systemctl restart named**

**4.在虚拟机A验证:nslookup www.qq.com 192.168.4.254**

**分离解析：**

**Split分离解析(视图解析)**

**• 当收到客户机的DNS查询请求的时候**

**– 能够区分客户机的来源地址**

**– 为不同类别的客户机提供不同的解析结果(IP地址)**

**– 针对不同的客户机就近提供服务器**

**BIND的view视图**

**• 根据源地址集合将客户机分类**

**– 不同客户机获得不同结果(待遇有差别)**

**– 由上及下进行匹配,匹配及停止**

**– 分类要合理,所有的客户端都要找到自己的分类**

**– 所有的zone都要在view中**

**虚拟机A:**

**[root@svr7 /]# vim /etc/named.conf**

**options {**

**directory "/var/named";**

**};**

**view "nsd" {**

**match-clients { 192.168.4.207; 192.168.7.0/24; }; #指定使用的DNS服务器客户端**

**zone "tedu.cn" IN {**

**type master;**

**file "tedu.cn.zone"; ---->解析结果 192.168.4.100**

**};**

**};**

**view "other" {**

**match-clients { any; };**

**zone "tedu.cn" IN {**

**type master;**

**file "tedu.cn.other"; ---->解析结果 1.2.3.4**

**};**

**};**

**多区域分离解析：**

**原则:每一个view中,zone的个数与负责的域名保持一致,地址库文件不同**

**[root@svr7 /]# vim /etc/named.conf**

**options {**

**directory "/var/named";**

**};**

**view "nsd" {**

**match-clients { 192.168.4.207; 192.168.7.0/24; };**

**zone "tedu.cn" IN {**

**type master;**

**file "tedu.cn.zone";**

**};**

**zone "qq.com" IN {**

**type master;**

**file "qq.com.zone"; ----> 解析结果 6.6.6.6**

**};**

**};**

**view "other" {**

**match-clients { any; };**

**zone "tedu.cn" IN {**

**type master;**

**file "tedu.cn.other";**

**};**

**zone "qq.com" IN {**

**type master;**

**file "qq.com.other"; ----> 解析结果 7.7.7.7**

**};**

**};**

**可以用acl简化地址集合**

**acl abc { 192.168.4.207; 192.168.7.0/24; };**

**view "nsd" {**

**match-clients { abc; };**

**zone "tedu.cn" IN {**

**type master;**

**file "tedu.cn.zone";**

**};**

**zone "qq.com" IN {**

**type master;**

**file "qq.com.zone";**

**};**

**};**

**RAID磁盘阵列：**

**廉价冗余磁盘阵列**

**– Redundant Arrays of Inexpensive Disks**

**• RAID 0,条带模式**

**– 同一个文档分散存放在不同磁盘**

**– 并行写入以提高效率**

**– 无可靠性提升**

**– 至少由两块磁盘组成**

**• RAID 1,镜像模式**

**– 一个文档复制成多份,分别写入不同磁盘**

**– 多份拷贝提高可靠性,效率无提升**

**– 至少由两块磁盘组成**

**• RAID5,高性价比模式**

**– 相当于RAID0和RAID1的折中方案**

**– 需要至少一块磁盘的容量来存放校验数据**

**– 至少由三块磁盘组成**

**• RAID6,高性价比/可靠模式**

**– 相当于扩展的RAID5阵列,提供2份独立校验方案**

**– 需要至少两块磁盘的容量来存放校验数据**

**– 至少由四块磁盘组成**

**• RAID 0+1/RAID 1+0 (RAID 10)**

**– 整合RAID 0、RAID 1的优势**

**– 并行存取提高效率、镜像写入提高可靠性**

**– 至少由四块磁盘组成**

**查看进程树**

**• pstree — Processes Tree**

**– 格式:pstree [选项] [PID或用户名]**

**• 常用命令选项**

**– -a:显示完整的命令行**

**– -p:列出对应PID编号**

**systemd:上帝进程(所有进程父进程) Linux初始化进程 PID为1**

**查看进程快照**

**• ps — Processes Snapshot**

**– 格式:ps [选项]...**

**• 常用命令选项**

**– aux:显示当前终端所有进程(a)、当前用户在所有**

**终端下的进程(x), 以用户格式输出(u)**

**– -elf:显示系统内所有进程(-e)、以长格式输出(-l)**

**信息、包括最完整的进程信息(-f)**

**• ps aux 操作**

**– 列出正在运行的所有进程,信息非常详细**

**• ps -elf 操作**

**– 列出正在运行的所有进程,会有PPID(父进程)信息**

**wc 统计**

**计算Linux一共有多少个用户**

**[root@svr7 /]# wc -l /etc/passwd**

**计算/etc/目录下有多少个.conf结尾的文件**

**[root@svr7 /]# find /etc -name "\*.conf" | wc -l**

**计算/etc/目录下有多少个tab结尾的文件**

**[root@svr7 /]# find /etc -name "\*tab" | wc -l**

**[root@svr7 /]# ps aux | wc -l #统计正在运行的进程有多少个**

**[root@svr7 /]# ps -elf | wc -l**

**[root@svr7 /]# ps aux #显示正在运行所有进程信息**

**[root@svr7 /]# ps -elf #显示正在运行进程信息,有PPID信息**

**进程动态排名**

**• top 交互式工具**

**– 格式:top [-d 刷新秒数] [-U 用户名]**

**按大写的P 利用cpu进行降序排列**

**按大写的M 利用内存使用比例,进行降序排列**

**检索进程**

**• pgrep — Process Grep**

**– 用途:pgrep [选项]... 查询条件**

**• 常用命令选项**

**– -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**

**[root@svr7 /]# pgrep a**

**[root@svr7 /]# pgrep -l log**

**[root@svr7 /]# pgrep -l cron**

**[root@svr7 /]# pgrep -lU lisi #查询lisi用户开启的进程**

**[root@svr7 /]# pstree -ap lisi**

**[root@svr7 /]# pgrep -lU lisi**

**[root@svr7 /]# who #查询当前登陆的用户信息**

**[root@svr7 /]# pgrep -lU lisi -t pts/2 #查询pts/2终端**

**[root@svr7 /]# top -d 1**

**#######################################################**

**进程的前后台调度**

**• 后台启动**

**– 在命令行末尾添加“&”符号,正在运行的放入后台**

**• Ctrl + z 组合键**

**– 挂起当前进程(暂停并转入后台)**

**• jobs 命令**

**– 查看后台任务列表**

**• fg 命令**

**– 将后台任务恢复到前台运行**

**• bg 命令**

**– 激活后台被挂起的任务**

**[root@svr7 /]# sleep 800 & #将进程正在运行放入后台**

**[root@svr7 /]# jobs #查看后台进程信息**

**[root@svr7 /]# sleep 700**

**^Z #按Ctrl+z暂停放入后台**

**[2]+ 已停止 sleep 700**

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

**[root@svr7 /]# bg 2 #将后台编号为2的进程,继续运行**

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

**[root@svr7 /]# fg 2 #将后台编号为2的进程,恢复到前台进行运行**

**sleep 700**

**^C #按Ctrl+c结束进程**

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

**[root@svr7 /]# fg 1 #将后台编号为1的进程,恢复到前台进行运行**

**sleep 800**

**^C #按Ctrl+c结束进程**

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

**####################################################3**

**杀死进程**

**• 干掉进程的不同方法**

**– Ctrl+c 组合键,中断当前命令程序**

**– kill [-9] PID... 、kill [-9] %后台任务编号**

**– killall [-9] 进程名...**

**– pkill 查找条件**

**[root@svr7 /]# sleep 800 &**

**[1] 30013**

**[root@svr7 /]# sleep 700 &**

**[2] 30014**

**[root@svr7 /]# sleep 600 &**

**[3] 30015**

**[root@svr7 /]# jobs -l**

**[root@svr7 /]# kill 30013**

**[root@svr7 /]# jobs -l**

**[root@svr7 /]# jobs -l**

**[root@svr7 /]# kill -9 30014**

**[root@svr7 /]# jobs -l**

**[root@svr7 /]# jobs -l**

**[root@svr7 /]# killall sleep**

**[root@svr7 /]# jobs -l**

**补充: 强制踢出一个用户(杀死该用户所有进程)**

**[root@svr7 /]# killall -9 -u lisi**

**日志概述**

**• 系统和程序的“日记本”**

**– 记录系统、程序运行中发生的各种事件**

**– 通过查看日志,了解及排除故障**

**– 信息安全控制的“依据”**

**• 常见的日志文件**

**/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**

**日志分析**

**• 通用分析工具**

**– tail、tailf、less、grep等文本浏览/检索命令**

**– awk、sed等格式化过滤工具**

**tailf:实时跟踪日志消息**

**用户登录分析**

**• users、who、w 命令**

**– 查看已登录的用户信息,详细度不同**

**• last、lastb 命令**

**– 查看最近登录成功/失败的用户信息**

**日志消息的优先级**

**• Linux内核定义的事件紧急程度**

**– 分为 0~7 共8种优先级别**

**– 其数值越小,表示对应事件越紧急/重要**

**0 EMERG（紧急） 会导致主机系统不可用的情况**

**1 ALERT（警告） 必须马上采取措施解决的问题**

**2 CRIT（严重） 比较严重的情况**

**3 ERR（错误） 运行出现错误**

**4 WARNING（提醒） 可能会影响系统功能的事件**

**5 NOTICE（注意） 不会影响系统但值得注意**

**6 INFO（信息） 一般信息**

**7 DEBUG（调试） 程序或系统调试信息等**

**一般超过四或者五 ggsimida**

**使用journalctl工具**

**• 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"**

**[root@svr7 /]# journalctl -u httpd #显示httpd服务日志**

**systemd介绍**

**• Linux系统和服务管理器**

**– 是内核引导之后加载的第一个初始化进程(PID=1)**

**– 负责掌控整个Linux的运行/服务资源组合**

**• 一个更高效的系统&服务管理器**

**– 开机服务并行启动,各系统服务间的精确依赖**

**– 配置目录:/etc/systemd/system/**

**– 服务目录:/lib/systemd/system/**

**– 主要管理工具:systemctl**

**对于服务的管理**

**systemctl restart 服务名 #重起服务**

**systemctl start 服务名 #开启服务**

**systemctl stop 服务名 #停止服务**

**systemctl status 服务名 #查看服务当前的状态**

**systemctl enable 服务名 #设置服务开机自启动**

**systemctl disable 服务名 #设置服务不开机自启动**

**RHEL7 运行模式**

**字符模式：multi-user.target**

**图形模式：graphical.target**

**[root@svr7 /]# ls -l /lib/systemd/system/**

**当前直接切换到图形模式**

**[root@svr7 /]# systemctl isolate graphical.target**

**当前直接切换到字符模式**

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

**查看每次开机默认进入模式**

**# systemctl get-default**

**multi-user.target**

**RHEL6 运行级别 200**

**0：关机 0**

**1：单用户模式（基本功能的实现，破解Linux密码） 50**

**2：多用户字符界面（不支持网络） 60**

**3：多用户字符界面（支持网络）服务器默认的运行级别 80**

**4：未定义**

**5：图形界面 160**

**6：重起 0**

**切换运行级别：init 5**

**设置永久策略，每次开机自动进入graphical.target**

**# systemctl set-default graphical.target**

**# reboot**

**PXE安装后脚本1：**

**useradd lisi**

**echo 123 | passwd --stdin lisi**

**echo 'begin-->IPSet' > /root/shell.txt**

**nmcli connection modify eth0 ipv4.method manual ipv4.addresses 192.168.4.22/24 connection.autoconnect yes**

**nmcli connection up eth0**

**if [ $? -eq 0 ];then**

**echo 'IP Success,Turn to yum ' >> /root/shell.txt**

**fi**

**echo '[pxe]**

**name=pxe**

**baseurl=ftp://192.168.4.254/rhel7**

**enabled=1**

**gpgcheck=0' > /etc/yum.repos.d/pxe.repo**

**yum repolist**

**echo 'yum Success' >> /root/shell.txt**

**PXE网络装机总结：**

**服务端**

1. **装dhcp包;**

**修改配置文件/etc/dhcp/dhcpd.conf**

**读入其他文件内容:r /usr/share/doc/dhcp\*/dhcpd.conf.example**

**subnet 192.168.4.0 netmask 255.255.255.0 { #指定分配网段**

**range 192.168.4.20 192.168.4.100; #指定分配IP地址范围**

**option domain-name-servers 8.8.8.8; #指定分配DNS服务器地址**

**option routers 192.168.4.254; #指定网关地址**

**default-lease-time 600;**

**max-lease-time 7200;**

**}**

**重启服务dhcpd**

1. **部署dhcp服务，同样修改配置文件 /etc/dhcp/dhcpd.conf**

**加入两行**

**subnet 192.168.4.0 netmask 255.255.255.0 {**

**range 192.168.4.20 192.168.4.100;**

**option domain-name-servers 8.8.8.8;**

**option routers 192.168.4.254;**

**default-lease-time 600;**

**max-lease-time 7200;**

**next-server 192.168.4.7; #指定下一个服务器IP地址**

**filename "pxelinux.0"; #指定网卡引导文件名称，记住pxelinux.0**

**}**

**重启服务dhcpd**

**（pxelinux.0:网络安装说明书,二进制文件）**

1. **搭建tftp服务 端口号 69,默认共享路径:/var/lib/tftpboot**

**装包 tftp-server**

**重启服务systemctl restart tftp**

**systemctl enable tftp ；**

**部署上面的pxelinux.0文件**

**yum provides \*/pxelinux.0 #查询仓库中哪个包产生此文件**

**yum -y install syslinux**

**rpm -ql syslinux | grep pxelinux.0 #查询安装清单**

**找到文件后cp到默认共享路径:/var/lib/tftpboot**

**cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/**

**ls /var/lib/tftpboot/**

**pxelinux.0中指明了默认的读取菜单文件的路径为：/var/lib/tftpboot/pxelinux.cfg/default**

1. **部署菜单文件**

**先挂载镜像以获得各类所需文件mount /dev/cdrom /mnt/**

**ls /mnt**

**复制所需文件，菜单文件isolinux/isolinux.cfg**

**mkdir /var/lib/tftpboot/pxelinux.cfg**

**cp /mnt/isolinux/isolinux.cfg /var/lib/tftpboot/pxelinux.cfg/default**

**给权限 chmod 644 /var/lib/tftpboot/pxelinux.cfg/default**

**5.背景图片splash.png 图形模块 vesamenu.c32**

**启动内核: vmlinuz**

**启动所需的驱动程序: initrd.img**

**cd /mnt/isolinux/**

**cp splash.png vesamenu.c32 vmlinuz initrd.img /var/lib/tftpboot/**

**ls /var/lib/tftpboot/**

**initrd.img pxelinux.0 pxelinux.cfg splash.png vesamenu.c32 vmlinuz (六份文件)**

1. **修改拷贝而来的菜单文件default：**

**vim /var/lib/tftpboot/pxelinux.cfg/default**

**1 default vesamenu.c32 #加载图形的模块**

**2 timeout 60 #读秒时间 1/10**

**........**

**10 menu background splash.png #背景图片**

**11 menu title NSD1811 PXE Server ! #修改标题**

**........**

**61 label linux**

**62 menu label Install RHEL7 #菜单显示内容**

**63 menu default #读秒结束默认选择**

**64 kernel vmlinuz #加载内核**

**65 append initrd=initrd.img #加载驱动程序**

1. **搭建web服务，提供安装树资源（RHEL7光盘目录结构）**

**yum -y install httpd**

**yum -y install httpd**

**[root@svr7 /]# systemctl restart httpd**

**[root@svr7 /]# ls /var/www/html/**

**[root@svr7 /]# firefox 192.168.4.7**

**创建挂载目录,进行挂载光驱设备**

**[root@svr7 /]# mount /dev/cdrom /var/www/html/rhel7**

**[root@svr7 /]# ls /var/www/html/rhel7**

**[root@svr7 /]# firefox 192.168.4.7/rhel7**

1. **部署无人值守安装，生成应答文件**

**yum -y install system-config-kickstart**

**修改yum仓库标识以使软件包选择 可用**

**im /etc/yum.repos.d/rhel7.repo**

**[development]**

**.........**

**设置system-config-kickstart 生成并且保存应答文件**

**利用web服务共享生成的应答文件ks.cfg**

**cp /root/ks.cfg /var/www/html/**

**[root@svr7 ~]# ls /var/www/html/**

**ks.cfg rhel7**

**[root@svr7 ~]# firefox 192.168.4.7/ks.cfg**

1. **再次修改菜单文件，指定ks应答文件位置**
2. **vim /var/lib/tftpboot/pxelinux.cfg/default**

**append initrd=initrd.img ks=http://192.168.4.7/ks.cfg**

**OK！**

**源码包编译安装之5步走：**

**步骤1:安装开发工具**

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

**步骤2: tar解包,释放源代码至指定目录**

**]# tar -xf /tools/inotify-tools-3.13.tar.gz -C /opt/**

**]# ls /opt/**

**1.txt inotify-tools-3.13**

**]# ls /opt/inotify-tools-3.13/**

**步骤3: ./configure 配置,指定安装目录/功能模块等选项**

**检测gcc是否安装**

**--prefix=指定安装路径**

**常见提示:未安装gcc软件**

**checking for gcc... no**

**checking for cc... no**

**checking for cl.exe... no**

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

**See `config.log' for more details.**

**]# cd /opt/inotify-tools-3.13/**

**]# ./configure**

**步骤4: make 编译,生成可执行的二进制程序文件**

**]# cd /opt/inotify-tools-3.13/**

**]# make**

**步骤5: make install 安装,将编译好的文件复制到安装目录**

**]# cd /opt/inotify-tools-3.13/**

**]# make install**

**]# ls /usr/local/bin/inotifywait #检测程序是否生成**

**/usr/local/bin/inotifywait**

**rsync同步操作：**

**rsync：只拷贝变化过的数据**

**• 命令用法**

**– rsync [选项...] 源目录 目标目录**

**• 同步与复制的差异**

**– 复制:完全拷贝源到目标**

**– 同步:增量拷贝,只传输变化过的数据**

**• rsync操作选项**

**– -n:测试同步过程,不做实际修改**

**– --delete:删除目标文件夹内多余的文档**

**– -a:归档模式,相当于-rlptgoD**

**– -v:显示详细操作信息**

**– -z:传输过程中启用压缩/解压**

**rsync -avz --delete**

**• 本地同步**

**– rsync [选项...] 本地目录1 本地目录2 //同步整个文件夹**

**– rsync [选项...] 本地目录1/ 本地目录2 //只同步目录下的数据**

**rsync+SSH同步**

**• 与远程的 SSH目录保持同步**

**– 下行:rsync [...] user@host:远程目录 本地目录**

**– 上行:rsync [...] 本地目录 user@host:远程目录**

**虚拟机svr7:**

**]# rsync -avz --delete /opt/ root@192.168.4.207:/opt/**

**虚拟机pc207:**

**]# ls /opt**

**两台服务器之间实时同步**

**一 部署公钥私钥ssh无密码验证**

**1.在虚拟机svr7生成公钥 私钥**

**[root@svr7 /]# ssh-keygen #一路回车**

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

**id\_rsa id\_rsa.pub known\_hosts**

**known\_hosts:记录曾经远程管理过的服务器信息**

**2.将公钥进行拷贝到虚拟机pc207**

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

**3.虚拟机pc207查看**

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

**authorized\_keys(公钥名字)**

**[root@pc207 /]#**

**4.虚拟机svr7验证无密码的同步**

**]# rsync -avz --delete /opt/ root@192.168.4.207:/opt/**

**二 运行一个程序,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**

**• 基本用法**

**– inotifywait [选项] 目标文件夹**

**• 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**

**[root@svr7 /]# inotifywait -mrq /opt/**

**三 书写一个shell脚本,完成实时同步**

**for循环适合写有次数的循环**

**while循环适合写死循环,当条件满足开始循环,条件失败停止循环**

**while [ 条件 ]**

**do**

**循环执行的操作**

**done**

**while 系统命令 #执行该命令,判断是否执行成功**

**do**

**循环执行的操作**

**done**

**[root@svr7 /]# vim /root/rsync.sh**

**#!/bin/bash**

**while inotifywait -rqq /opt/**

**do**

**rsync -az --delete /opt/ root@192.168.4.207:/opt/**

**done**

**[root@svr7 /]# chmod +x /root/rsync.sh**

**[root@svr7 /]# /root/rsync.sh**

**[root@svr7 /]# /root/rsync.sh &**

**[1] 30381**

**[root@svr7 /]# jobs #查看后台运行的脚本信息**

**[1]+ 运行中 /root/rsync.sh &**

**[root@svr7 /]# killall rsync.sh #杀死运行的脚本**

**环境变量:**

**USER=当前系统登陆的用户名**

**PATH=储存命令程序的搜寻的路径**

**对于PATH**

**可以利用PATH的路径如/usr/bin 进行程序的执行。**

**[root@svr7 ~]# echo $PATH**

**/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin**

**[root@svr7 ~]# vim /opt/hello.sh**

**#!/bin/bash**

**echo hahaxixi**

**[root@svr7 ~]# chmod +x /opt/hello.sh**

**[root@svr7 ~]# /opt/hello.sh**

**[root@svr7 ~]# hello.sh #执行失败**

**[root@svr7 ~]# cp /opt/hello.sh /usr/bin**

**[root@svr7 ~]# hello.sh #执行成功**

**书写周期性计划任务,命令写绝对路径**

**查询命令对应程序文件在哪里?**

**[root@svr7 /]# which 命令**

**[root@svr7 /]# which poweroff**

**/usr/sbin/poweroff**

**[root@svr7 /]#**

**######################################################**

**权限掩码**

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

**0022**

**[root@svr7 /]# umask 077 #改为0077**

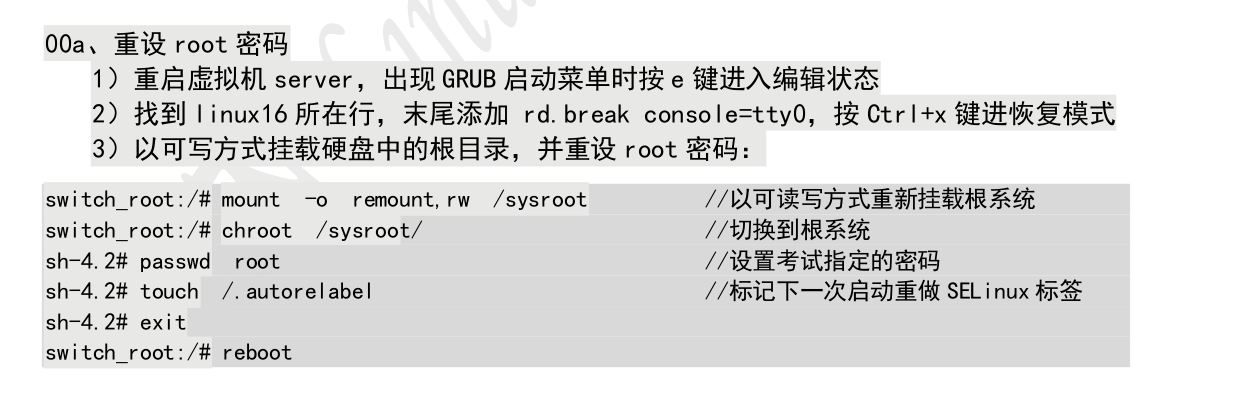
**[root@svr7 /]# umask**

**0077**

**[root@svr7 /]# mkdir /opt/nsd01**

**[root@svr7 /]# ls -ld /opt/nsd01**

**drwx------. 2 root root 6 12月 25 09:35 /opt/nsd01**

****

**（发送请求）三次握手**

**客户端 SYN ---- 服务端 ACK，SYN -----客户端 ACK**

**四次断开**

**客户端 FIN ---- 服务端 ACK --- 服务端 FIN ---客户端 ACK**

**记住开启vlan**

**vlan:限制广播，增加安全，提高带宽利用率**

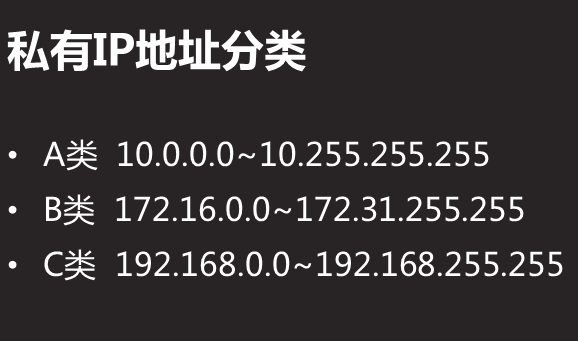
**交换机是组建局域网的设备，工作在数据链路层，传播数据帧，根据MAC地址转发数据.**

**32-n(n为掩码块) 个0**

**imap tcp 143**

**tcp 可靠性高,面向连接 效率低**

**udp 不可靠,无连接 效率高**

****