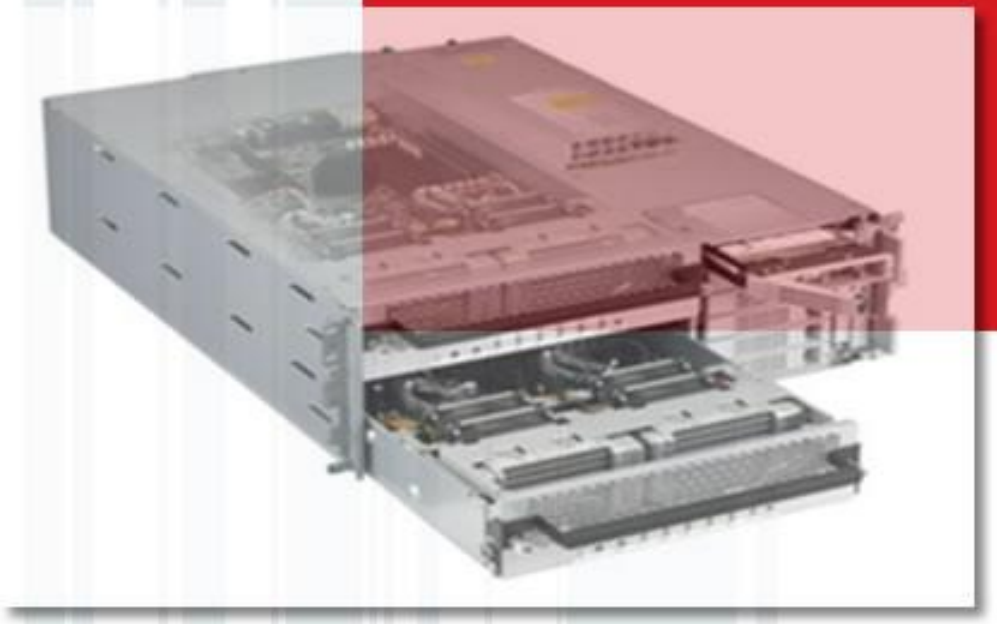
Linux 系统管理

DSC认证培训体系



课程时间：120分钟

更新日期：2008年11月

HPCTC
高性能计算专业委员会
High Performance Computing Technical Committee

曙光 DAWNING

课程简介

- 本课程为**Linux操作系统**培训课程之一
- 课程内容：Linux操作系统的系统管理
- 培训对象：
  - 参加曙光DCSA认证的技术人员
- 能力要求：深入了解Linux操作系统的相关配置及管理
- 培养目标：此培训纲要针对Linux中级学习，使大家能够独立安装配置及管理Linux；掌握Linux操作系统的相关管理知识及一些常用的服务配置。

目 录

第一章　系统用户及工作组管理

- 授课内容
  - 1、用户及工作组的简介
  - 2、用户的管理
  - 3、工作组的管理
- 授课目标
  - 1、熟悉用户的创建和管理
  - 2、熟悉工作组的创建和管理

用户管理(一)

帐号管理

帐号建立:

- 用useradd命令可以进行帐号建立新使用者的信息。
- 语法: useradd 参数 username

常用参数:

-c : comment 指定一段注释性描述。

-d: 目录 指定用户主目录

-e: 帐号终止日期。日期的指定格式为 MM/DD/YY

-g: 指定用户所属的用户组

-G: 指定用户所属的附加组

-m: 能够创建主目录

-p: 设置用户密码

-u: 指定用户的用户ID号

-f :帐号过期几日后永久停权。当值为0时帐号则立刻被停权。而当值为-1时则关闭此功能，预设值为-1

用户管理(二)

帐号管理

- 修改使用者帐号:

  用命令usermod修改使用者帐号。

  语法: usermod 参数 username

参数:

```
usermod -d 目录 用户名 #修改用户目录
usermod -g 组名 用户名 #修改用户主要组
usermod -G 组名 用户名 #修改用户附属组
usermod -L 用户名 #锁定用户
usermod -U 用户名 #解锁用户
usermod -l 新名 旧名 #修改用户名
```

用户管理（三）

- 密码的设置passwd
- 语法：passwd 参数 用户名
- 参数：

```
passwd -d 用户名      #清空密码
passwd -l 用户名      #锁定用户，该用户不能够使用
passwd -u 用户名      #解锁用户
```

用户管理（四）

- 用户的删除

语法：userdel 参数 用户名

参数：

userdel -r 用户名 #连同目录一起删除

用户管理文件

- /etc/passwd
  - root:x:0:0:root:/root:/bin/bash
  - bin:x:1:1:bin:/bin:/sbin/nologin
  - 用户名:加密的口令:用户ID:组ID:用户的全名或描述:登录目录:登录shell
- /etc/shadow
  - root:1xxr3zJpz$kq7UpH12t7KkSEdNrfBK1/:12863:0:99999:7:::
  - 用户登录名
  - 用户加密后的口令，(若为空表示改用户不需口令即可登陆，若为*号，表示帐号被禁止)
  - 从1970年1月1日至口令最近一次被修改的天数
  - 口令在多少天内不能被用户修改
  - 口令在多少天后必须被修改
  - 口令到期前多少天开始给用户发出警告
  - 口令过期多少天后用户帐号被禁止
  - 自1970年1月1日到帐号过期那一天的天数
  - 保留域

组管理（一）

用户组管理

建立新组

用groupadd命令来建立新群组。

语法：groupadd　参数　组名

参数：

groupadd -g gid号　组名　#gid大于500，gid自设

groupadd -r gid　组名　#gid小于500，gid自设

修改组

用groupmod命令来修改群组

语法：groupmod 参数　组名

参数：

组管理（二）

- 组的删除

  语法：groupdel　组名

工作组管理文件

- /etc/group
  - root:x:0:root
  - bin:x:1:root,bin,daemon
  - sys用户组:设有口令:组ID为3:组成员有root，bin，adm
- /etc/gshadow
  - root:::root
  - bin:::root,bin,daemon
  - 组名:组加密密码:组管理:组成员

其他相关配置文件

- /etc/default/useradd

  # useradd defaults file
  GROUP=100　注：预设的组ID
  HOME=/home　注：把用户的家目录建在/home中；
  INACTIVE=-1 注：是否启用帐号过期停权，-1表示不启用；
  EXPIRE=　　注：帐号终止日期，不设置表示不启用；
  SHELL=/bin/bash 注：所用SHELL的类型；
  SKEL=/etc/skel 注：　默认添加用户的目录默认文件存放位置；也就是说，当我们用adduser添加用户时，用户家目录下的文件，都是从这个目录中复制过去的；
- useradd的默认设置文件：**useradd －D**

其他相关配置文件

- **/etc/login.defs 配置文件**
  /etc/login.defs 文件是当创建用户时的一些规划，比如创建用户时，是否需要家目录，UID和GID的范围；用户的期限等等，这个文件是可以通过root来定义的；

  比如Redhat的 /etc/logins.defs 文件内容（注释删除后的内容）

  MAIL_DIR /var/spool/mail 注：创建用户时，要在目录/var/spool/mail中创建一个用户mail文件；
  PASS_MAX_DAYS 99999 注：用户的密码不过期最多的天数；
  PASS_MIN_DAYS 0 注：密码修改之间最小的天数；
  PASS_MIN_LEN 5 注：密码最小长度；
  PASS_WARN_AGE 7 注：密码失效前几天提醒用户
  UID_MIN 500 注：最小UID从500开始；
  UID_MAX 60000 注：最大UID到60000结束；
  CREATE_HOME yes 注：是否创用户家目录，要求创建；

用户权限管理

- **默认权限分配的命令 umask**

  umask 是通过八进制的数值来定义用户创建文件或目录的默认权限,umask 表示的是禁止权限,不过文件和目录有点不同:

  新创建的文件默认不具有可执行允可权限：-rw-------

  新创建的目录默认具有可执行允可权限： drwx--x--x

  对于文件来说，umask 的设置是在假定文件拥有八进制666权限上进行，文件的权限就是是666减去umask的掩码数值;
  对于目录来说，umask 的设置是在假定文件拥有八进制777权限上进行，目录八进制权限777减去umask的掩码数值

系统用户的家目录的权限是通过在配置文件中指定的，比如Fedora 中是用的 /etc/login.defs文件；其中有这样一段:
CREATE_HOME yes
UMASK 077
表示的意思是，当我们创建用户时，他的家目录umask的数值是077 。我们怎么理解这个077呢。
当用户添加时，系统自动在/home中创建用户的家目录，并且设置它的权限为777-077=700 ，也就是rwx------

umask	文件	目录
0	6	7
1	5	6
2	4	5
3	3	4
4	2	3
5	1	2
6	0	1
7	0	0

用户管理实例

- 添加test用户,该用户属于workgroup组

  添加组:

  #groupadd -u 1000 workgroup

  添加用户:

  #useradd -u 1000 -g 1000 -d /home/test -s /bin/bash test

  若系统为suse，则要在添加用户时生成用户目录，需要加-m参数

  #useradd -u 1000 -g 1000 -m /home/test -s /bin/bash test
- 删除test用户

  #userdel test

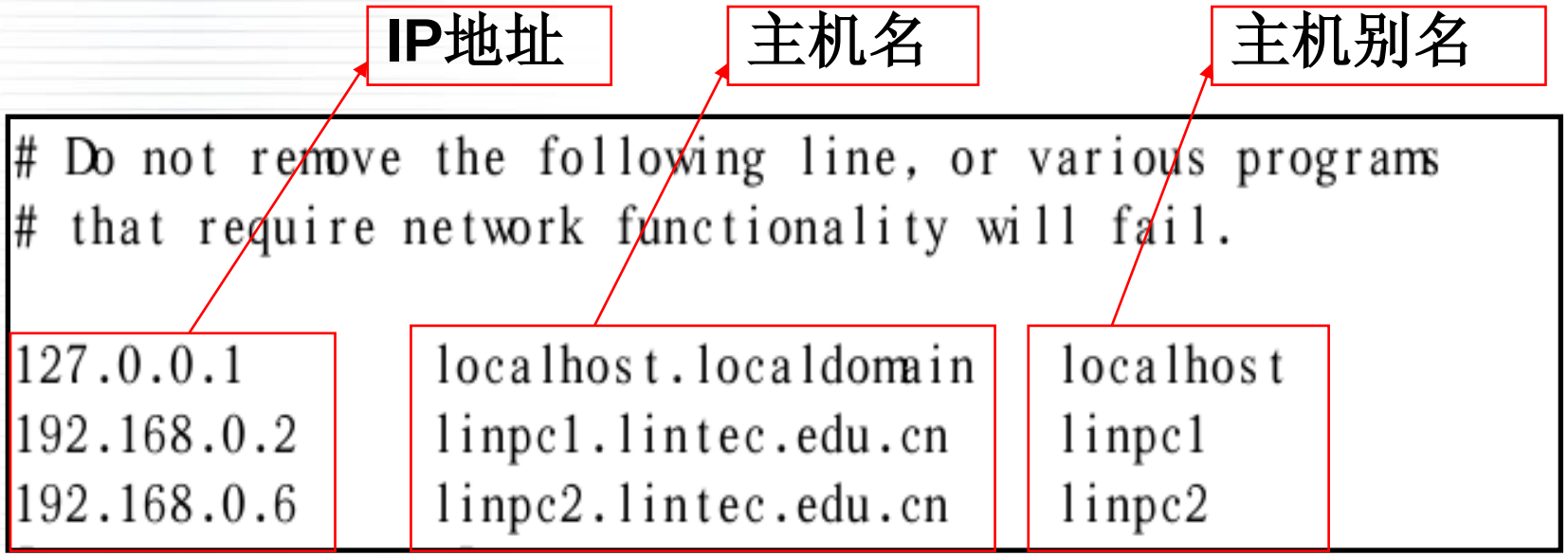
  #rm -rf /home/test

目 录

第二章　系统网络管理

网络配置文件管理（一）

- /etc/hosts
  网络映射文件



网络配置文件管理（二）

- /etc/services

  端口映射文件

```
# The latest IANA port assignments can be gotten from
#       http://www.iana.org/assignments/port-numbers
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# Each line describes one service, and is of the form:
#
# service-name  port/protocol  [aliases ...]   [# comment]

tcpmux          1/tcp                                   # TCP port service multiplexer
tcpmux          1/udp                                   # TCP port service multiplexer
rje             5/tcp                                   # Remote Job Entry
rje             5/udp                                   # Remote Job Entry
```

服务名　端口号　协议

网络配置文件管理（三）

- /etc/sysconfig/network
  系统名字配置文件

NETWORKING=YES|NO	**YES** 表示需要配置网络； **NO** 表示不需要配置网络；
HOSTNAME=hostname	主机的全限定域名；
GATEWAY=gw-ip	网络网关的 **IP** 地址；
GATEWAYDEV=gw-dev	网关设备的名称（例如 **eth0**或**IP**地址）；
NISDOMAIN=dom-name	表示 **NIS**域(网络信息服务，用来在一个域共享用户、密码等信息，类似于**Windows**域服务）。

网络配置文件管理（四）

- /etc/host.conf
- 当系统中同时存在DNS域名解析和/etc/hosts主机表机制时，由该/etc/host.conf确定主机名解释顺序。
- 示例：

  order hosts,bind #名称解释顺序

  multi on #允许主机拥有多个IP地址

  nospoof on #禁止IP地址欺骗

  order是关键字，定义先用本机hosts主机表进行名称解释，如果不能解释，再搜索bind名称服务器(DNS)。

设置名字查找顺序

```
[root@linpc1 root]# cat /etc/host.conf
order hosts,bind
multi on
```

一个主机名可以有多个地址

网络配置文件管理（五）

- /etc/nsswitch.conf
  nsswitch.conf文件不仅能处理主机表和DNS之间的优先次序，还能处理其它的问题。它为几个不同的系统管理数据库定义来源

```
passwd:         files
shadow:         files
group:          files

#hosts:         db files nisplus nis dns
hosts:          files dns
```

在hosts行加入dns解析方法，
表明通过主机列表文件
和dns查找主机名

网络配置文件管理（六）

- /etc/resolv.conf

  文件/etc/resolv.conf用于配置DNS客户端，它包含了主机的域名搜索顺序和DNS服务器的地址，每一行应包含一个关键字和一个或多个的由空格隔开的参数

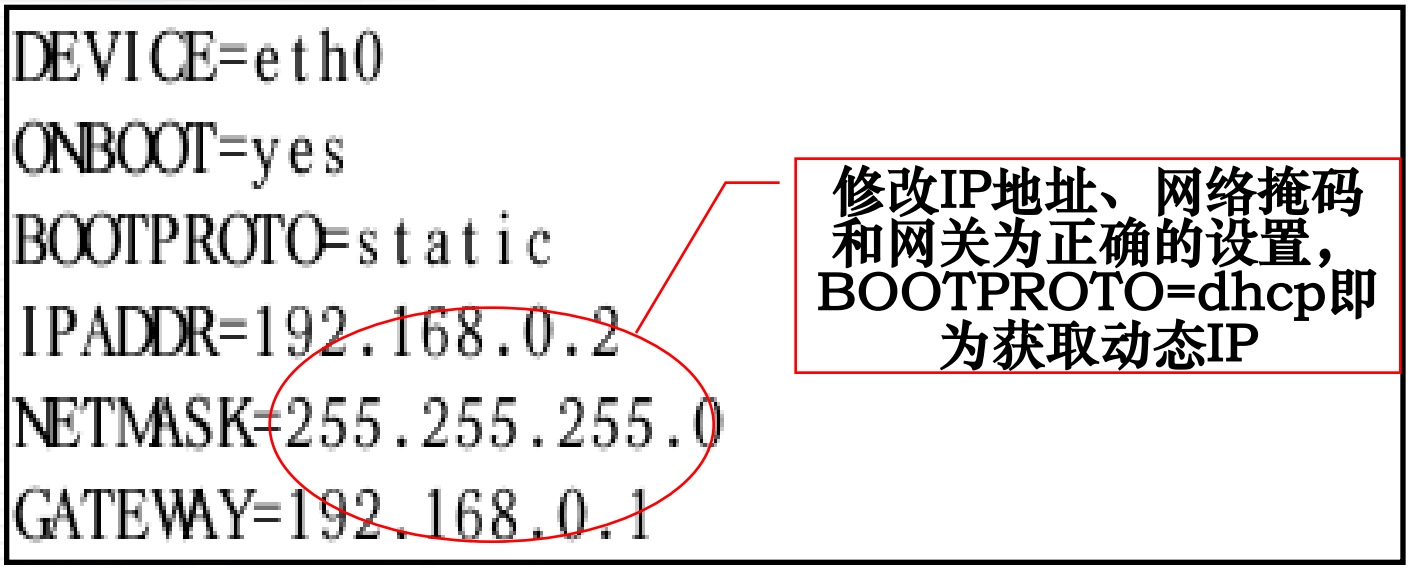
```
[root@linpc1 root]# cat /etc/resolv.conf
nameserver 202.10.0.20
search lintec.edu.cn
```

名字服务器的地址

在搜索不带分割点的名称时，在该名字的后面加上search域的值，作为该名字的全称域名

网络配置文件管理（七）

- /etc/sysconfig/network-scripts/ifcfg-interface-name
  interface-name： eth0,eth1…



网络管理命令（一）

- ifconfig

观察网卡MAC地址及IP地址

- ifconfig eth0 down

关闭网络IP

- ifconfig eth0 up

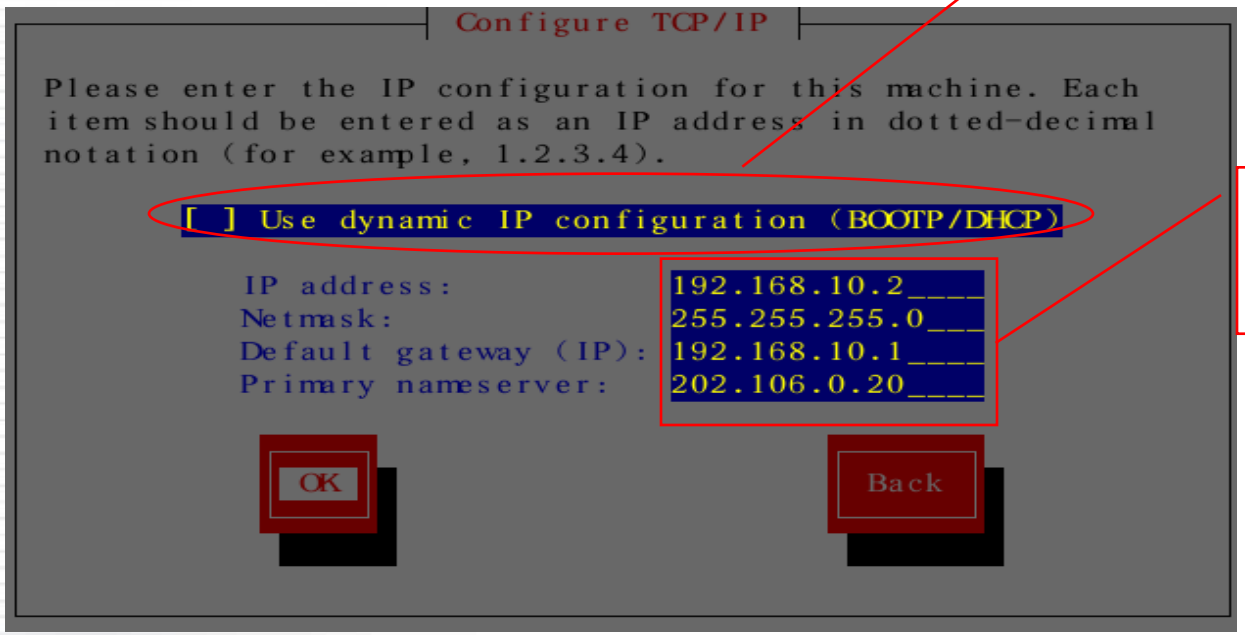
启动网络IP

- ifconfig eth0:0 192.168.0.1 netmask 255.255.255.0

在第一块网卡上绑定一个临时IP地址

网络管理命令（二）

- netconfig(neat)
  启动网络配置界面



DHCP设置选项

静态地址及网关、掩码，DNS设置

/etc/init.d/network restart

网络管理命令（三）

- ping

  网络测试命令

```
[root@linpc1 root]# ping -c 4 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=0.302 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=0.159 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=0.153 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=0.152 ms

--- 192.168.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 0.152/0.191/0.302/0.065 ms
```

网络管理命令（四）

- route
  - 添加路由

```
route add -net 192.168.80.0 netmask 255.255.255.0 gw 192.168.0.80
```

  - 查看路由

```
[root@linpc1 root]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.80.0    192.168.0.80    255.255.255.0   UG    0      0        0 eth0
192.168.0.0     *               255.255.255.0   U     0      0        0 eth0
169.254.0.0     *               255.255.0.0     U     0      0        0 eth0
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         192.168.0.1     0.0.0.0         UG    0      0        0 eth0
```

  - 删除路由

```
[root@linpc1 root]# route del -net 192.168.80.0 netmask 255.255.255.0
```

网络管理命令（五）

- traceroute
  - 语法

```
Version 1.4a12
Usage: traceroute [-dFInrvx] [-g gateway] [-i iface] [-f first_ttl]
        [-m max_ttl] [ -p port] [-q nqueries] [-s src_addr] [-t tos]
        [-w waittime] [-z pausemsecs] host [packetlen]
```

  - 查找到达目的主机的路由路径

```
[root@linpc1 root]# traceroute www.sina.com.cn
traceroute: Warning: www.sina.com.cn has multiple addresses; using 202.106.185.250
traceroute to libra.sina.com.cn (202.106.185.250), 30 hops max, 38 byte packets
 1  61.148.8.50 (61.148.8.50)  8.044 ms  11.074 ms  2.592 ms
 2  61.148.8.49 (61.148.8.49)  1.951 ms  2.235 ms  1.748 ms
 3  61.148.4.9 (61.148.4.9)  1.125 ms  1.226 ms  1.100 ms
 4  202.108.46.29 (202.108.46.29)  1.093 ms  1.115 ms  0.913 ms
 5  202.106.193.149 (202.10...  1.213 ms  1.335 ms  1.410 ms
 6  RTR...bta.net.cn (...192.170)  1.200 ms
 7  210... (210.74....876 ms  1.76...
 8  202.106.185.250 (202.106.185.250)  5.982 ms  1.757 ms  1.156 ms
```

第二个网关

网关的IP地址

到达该网关所需的时间

目 录

第三章 系统磁盘管理

磁盘限额简介

- 磁盘限额
  - 用户可以在特定的文件系统上进行磁盘限额操作，可以设定磁盘块的数量或inodes的数量
  - 硬限制和软限制
    - 软限制：文件占用磁盘容量可以超出软限制值，系统会提示超出，但以硬限制为边界
    - 硬限制：文件占用磁盘容量的边界
    - 磁盘配额可以设置为on或off状态

Linux系统磁盘限额的设置

- 确定磁盘限额的目标磁盘，举例如下
  [root@node1 ~]#mount /dev/sdb1 /home ，对磁盘/dev/sdb1做限额
- 文件/etc/fstab 的修改
  在/etc/fstab 中添加参数，开启文件系统的配额设置。对于用户，添加 usrquota ；对于组，添加grpquota
  [root@node1 ~]#vi /etc/fstab
  /dev/sdb1 /home ext3 defaults,usrquota,grpquota 0 0
- 修改文件/etc/fstab后，使其生效
  [root@node1 ~]#mount -o remount /dev/sdb1
  然后，执行mount 命令，请确认已经显示出usrquota 和usrquota 文字。
  [root@node1 ~]# mount
  显示包含
  /dev/sdb1 on /home type ext3 (rw,usrquota,grpquota)

Linux系统磁盘限额的设置

•aquota.user 、aquota.group 文件的制作

设置磁盘限额时，必须事前制作用户磁盘限额的配置文aquota.user和组磁盘限额的配置文件aqouta.group。

制作aquota.user,aquota.group文件时，执行如下命令：

[root@localhost home]#touch aquota.user

[root@localhost home]#touch aquota.group

[root@localhost home]# quotacheck －m(强制) -u /home

若磁盘容量比较大，比如达到了几个T，则该命令要执行将近半个小时左右

[root@localhost home]# quotacheck -g /home

磁盘限额相关的命令加上选项-u 时含义为用户磁盘限额，加上选项-g时含义为组磁盘限额。什么都不加时缺省为用户磁盘限额。

执行命令时，在将磁盘限额配置文件放在有效的文件系统的路径(本例为/home)中。

Linux系统磁盘限额的设置

- 编辑磁盘限额配置文件

使用edquota 进行编辑 (也可以用setquota －u <user> <soft block> <hard block> <soft inode> <hard inode> <filesystem> 完成)

- 举例:

要对用户test 进行的磁盘限额设置时，执行如下命令。

[root@localhost home]# edquota -u test

启动编辑程序，进入后如下所示:

Disk quotas for user test (uid500):

Filesustem	blocks	soft	hard	inodes	soft	hard
/dev/sdb1	默认设置	(以K为单位)	(以K为单位)	默认设置	文件目录个数	

blocks 表示用户在该分区上已经消费的块数。第一个soft和hard即是要限制的容量，以K为单位，inodes 表示已经使用的i节点数。第二个soft和hard用来限制该用户占用磁盘空间中的文件目录个数。

以上参数如果以0表示，则为不限制

[root@localhost home]# quota －u test 查看用户限制情况

对组workgroup的磁盘限额进行设置时，执行如下命令。

[root@localhost home]# edquota -g workgroup

Linux系统磁盘限额的设置

- 磁盘限额配置完成后，需要启动限额服务，使配置生效

[root@localhost home]#quotaon –avug

失效命令：

[root@localhost home]#quotaoff –avug

- 设置用户磁盘限额的宽限期

[root@localhost home]# edquota -u -t

grace period before enforcing soft limits for users:

Time units may be: days, hours, minutes, or seconds

Filesystem	Block grace period	Inode grace period
	7 days	7 days

block grace period 表示对程序块数的宽限期

inode grace period 表示对i节点的宽限期

单位可以使用日(days)、小时(hours)、分(minutes)、秒(seconds)

该限制在用户文件容量超过soft限制，7天后，磁盘不能够再读写

- 批量用户磁盘限额

很多用户的磁盘限额大小及配置均相同的前提下，可以使用下列命令：

[root@localhost home]# edquota -p test test1 test2 test3 …来实现

目 录

第四章 系统SAMBA管理

- SMB协议和Samba简介
- 安装和启动Samba
- 配置Samba文件共享
- 在Linux环境下访问Samba共享

SMB协议

- SMB（Server Message Block，服务信息块）协议是一个高层协议， 它提供了在网络上的不同计算机之间共享文件、打印机和不同通信资料的手段。
- SMB的工作原理就是让 NetBIOS 与 SMB 协议运行在TCP/IP上， 并且使用NetBIOS的名字解释器让Linux机器可以在 Windows 的网上邻居中被看到， 从而和Windows9X/NT 进行相互沟通，共享文件和打印机。

Sabma简介

- Samba是一组软件包，使Linux支持SMB协议，该协议是在TCP/IP上实现的， 它是Windows网络文件和打印共享的基础，负责处理和使用远程文件和资源。
- Samba的核心是两个守护进程

  smbd：监听139 TCP端口 ，处理到来的SMB数据包。

  nmbd：监听137和138 UDP端口 ，使其它主机（或工作站）能浏览Linux服务器 。

Sabma简介

- Samba软件的功能
  - 共享Linux的文件系统。
  - 共享安装在Samba服务器上的打印机。
  - 支持Windows客户使用网上邻居浏览网络。
  - 使用Windows系统共享的文件和打印机。
  - 支持Windows域控制器和Windows成员服务器对使用Samba资源的用户进行认证。
  - 支持WINS名字服务器解析及浏览。
  - 支持SSL安全套接层协议。

服务器的安装和启动

- Samba服务器的安装
  - # rpm -ivh samba-common-*.rpm
  - # rpm -ivh samba-*.rpm
  - # rpm -ivh samba-client-*.rpm
- Samba的启动和停止
  - # service smb start
  - # service smb stop
  - # service smb restart
- Samba的配置文件：/etc/samba/smb.conf

Samba的默认配置

- 工作组：MYGROUP
- 安全等级：user
- 设置用户密码加密：Yes
- 口令文件路径：/etc/samba/smbpasswd
- 认证用户时服从PAM的管理限制：Yes
- 为客户做DNS查询：No
- 设置了每个用户的主目录的共享
- 设置了全部打印机的共享

建立Samba口令文件设置Samba账号

- 设置现有用户与smb用户的一致性

  cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd

- 添加单个的samba账户

  # smbpasswd -a username

测试的默认配置

- 检测Samba配置文件的正确性
  - # testparm
- 在Windows上访问Samba资源共享
  - 通过Windows的网上邻居访问Samba共享
  - 通过映射网络驱动器访问Samba共享
- 在Linux服务器上列出Samba的资源使用情况
  - # smbstatus

Samba配置基础

- smb.conf文件的分节结构
  - [Global]：用于定义全局参数和缺省值
  - [Homes]：用于定义用户的Home目录共享
  - [Printers]：用于定义打印机共享
  - [Userdefined_ShareName]：用户自定义共享（可有多个）
- Samba的安全等级
  - Share：用户不需要账户及口令即可登入Samba服务器。
  - User：由提供服务的Samba服务器负责检查账户及口令（是Samba默认的安全等级）。
  - Server：检查账户及口令的工作指定由另一台Windows NT/2000或Samba服务器负责。
  - Domain：指定Windows NT/2000域控制服务器来验证用户的账户及口令。

设置Samba的全局参数（1）

- 基本全局参数
  - netbios name：设置Samba的NetBIOS名字
  - workgroup：设置Samba要加入的工作组
  - server string：指定浏览列表里的机器描述
  - client code page：设置客户字符编码页
- 日志全局参数
  - log file：指定日志文件的名称
  - max log size：指定日志文件的最大尺寸（KB）

设置Samba的全局参（2）

- 安全全局参数
  - socket address：指定samba监听的IP地址
  - admin user：设置管理员账号
  - security：定义Samba的安全级别
  - encrypt passwords：用于指定是否使用加密口令
  - smb passwd file：指定Samba口令文件的路径
  - hosts allow：指定可以访问Samba的主机
  - hosts deny：指定不可以访问Samba的主机

设置Samba的全局参（3）

- 运行效率全局参数
  - change notify timeout：设置服务器周期性异常通知
  - deadtime：客户端无操作多少分钟后服务器端中断连接
  - getwd cache：是否使用Cache功能
  - keepalive：服务器每隔多少秒向客户端发送keepalive包用于确认客户端是否工作正常
  - max open files：同一个客户端最多能打开的文件数目
  - socket options：设置服务器和客户之间会话的Socket选项

设置Samba共享资源参数

- 基本共享参数
  - comment：指定对共享的描述
  - path：指定共享服务的路径
- 访问控制参数
  - writable：指定共享的路径是否可写
  - browseable：指定共享的路径是否可浏览（默认为可以）
  - available：指定共享资源是否可用
  - read only：指定共享的路径是否为只读
  - public：指定是否可以允许guest账户访问
  - read list：设置只读访问用户列表
  - write list：设置读写访问用户列表
  - valid users：指定允许使用服务的用户列表
  - invalid users：指定不允许使用服务的用户列表

文件系统权限和Samba共享权限

- Samba服务器要将本地文件系统共享给Samba用户，涉及两种权限：
  - 本机文件系统权限：使用chmod和chown命令设置
  - Samba 权限：使用Samba的访问控制参数设置
- 当Samba用户访问共享时，最终的权限将是这两种权限中最严格的权限。

在Linux环境下访问Samba共享（1）

- lmhosts文件
  - Samba使用/etc/samba/lmhosts文件存放NetBIOS名与IP地址的静态映射表
- smbclient 命令
  - Samba提供了一个类似FTP客户程序的Samba客户程序smbclient
  - 可以使用smbclient查看并访问共享
- 列表显示指定主机提供的共享
  - # smbclient -L NetBIOS名或IP地址
  - 例如:
- # smbclient -L win01

使用smbclient命令访问共享（2）

- Samba提供了一个类似FTP客户程序的Samba客户程序smbclient
- 用于访问指定主机的指定共享，-U 用户名参数表示以指定的用户名的身份访问共享。
  - # smbclient //NetBIOS名或IP地址/共享名 -U 用户名
  - 注意：
    - 当访问Windows共享时，smbclient命令的 -U参数后所指定的用户名是所访问的Windows计算机中的用户账户， 验证口令是Windows计算机中的用户账户的口令。
    - 当访问Linux提供的Samba共享时，smbclient命令的 -U参数后所指定的用户名是所访问的Linux计算机中的Samba用户账户，验证口令是Samba用户账户的口令。
  - 例如：
- # smbclient //win01/tools -U osmond

在Linux环境下访问Samba共享（3）

- 使用smbmount挂装远程SMB文件系统访问Samba共享
  - smbmount命令格式
- # smbmount //NetBIOS名或IP地址/共享名 挂装点
  - 例如：

  # smbmount //win01/tools /mnt/smb/win01
- 卸载：

  # umount /mnt/smb/win01

Samba服务配置举例

- 增加samba共享用户

  useradd smbuser

  passwd　smbuser

  smbpasswd －a smbuser

Samba服务配置举例

- 编辑/etc/smb.conf这个文件

```
workgroup = MSHOME
 netbios name = smbserver(网络名字)
 server string  = Samba server（网络描述）
 security = user     （安全级别）
 encrypt passwords = yes    （密码加密）
 smb passwd file = /etc/samba/smbpasswd
 [smbuser]
     comment = smbuser
     path = /home/smbuser
     valid users = smbuser
     browsenable = yes
     writable = yes
```

目 录

第五章 系统NFS管理

- NFS系统简介
- NFS服务的安装与配置
- NFS服务测试

NFS系统简介

- NFS (Network file system) 最初是由 Sun 公司于 1984 年开发出来的，最主要的功能就是让网络上的 UNIX 电脑可以共享目录及档案。我们可以将远端所分享出来的档案系统，挂载 (mount) 在本地端的系统上，然后就可以很方便的使用远端的档案，而操作起来就像在本地操作一样，不会感到有甚么不同。
- NFS server可以看作是一个FILE SERVER

NFS系统简介

- RPC（Remote Procedure Call)

  虽然 NFS 有属于自己的协议和所使用的 port number，但NFS 本身没有提供信息传输的协议和功能。NFS之所以能让我们通过网络进行资料的分享，这是因为NFS使用了一些其它的传输协议，而这些传输协议用到这个RPC功能。

NFS系统简介

➢RPC（Remote Procedure Call)

可以将NFS服务器看成是一个RPC服务器，NFS客户端看成是一个RPC客户端，所以只要用到NFS的地方都要启动RPC服务，不论是NFS SERVER或者NFS CLIENT。这样SERVER和CLIENT才能通过RPC来实现PROGRAM PORT的对应。可以这么理解RPC和NFS的关系：NFS是一个文件系统，而RPC是负责负责信息的传输。

NFS系统简介

➢举个例子来说：当 Client 端尝试来使用 RPC server 所提供的服务时，由于 Client 需要取得一个可以连接的 port 才能够使用 RPC server 所提供的服务，因此， Client 首先就会去跟 portmap 讲『喂！可不可以通知一下，给我个 port number，好让我可以跟 RPC 联络吧！』这个时候 portmap 就自动的将自己管理的 port mapping 告知 Client，好让他可以连接上来 server 呢！所以啰：『激活 NFS 之前，请先激活 portmap！』

NFS系统简介

- NFS需要启动的DAEMONS
  - rpc.nfsd:主要负责登陆权限检测。
  - rpc.mountd：主要功能是管理NFS的文件系统。当客户端顺利的通过rpc.nfsd登陆NFS服务器后，在使用NFS服务器所提供的文件前，还必须通过文件使用权限的验证， rpc.mountd会读取NFS的配置文件/etc/exports来对比客户端的权限。
  - portmap: 主要功能是进行端口映射工作。当客户端尝试连接并使用RPC服务提供的服务试，portmap会将所管理的与服务对应的端口号提供给客户端，从而使客户端可以通过端口向服务器请求服务

- 注：在REDHAT中PORTMAP是默认启动的

NFS服务的安装

- NFS SERVER在REDHAT LINUX平台下一共需要两个软件包：nfs-utils和portmap
- nfs-utils：提供rpc.nfsd 及 rpc.mountd这两个NFS DAEMONS的软件
- portmap: NFS其实可以被看作是一个RPC SERVER PROGRAM,而要启动一个RPC SERVER PROGRAM，都要做好PORT的对应工作，而且这样的任务就是由PORTMAP来完成的。通俗的说PortMap就是用来做PORT的mapping的。

NFS服务的配置

目前的所有Linux操作系统，默认均安装了NFS服务，但服务未开启，配置前需要打开该服务

配置前检查是否开启了portmap、nfs、nfslock服务

[root@node1~]#chkconfig nfs on

[root@node1~]#/etc/init.d/nfs start

配置文件/etc/exports的编辑

[root@node1 ~]#vi /etc/exports

/public *(rw,no_root_squash,async)

参数说明

ro read only

rw read write

no_root_squash 信任客户端，对应 UID

*可以设置为IP、HOSTNAME等，限制客户端的IP范围

配置文件生效

[root@node1 ~]#exportfs -a

[root@node1 ~]#exportfs

/public <world>

说明nfs服务端设置生效，可以在客户端执行mount服务端加载共享服务

NFS服务的配置

- 可以设定的参数主要有以下这些：

rw：可读写的权限；
ro：只读的权限；
no_root_squash：登入到NFS主机的用户如果是ROOT用户，他就拥有ROOT的权限，此参数很不安全，建议不要使用。
root_squash: 登入 NFS 主机使用分享目录的使用者如果是 root 时，那么这个使用者的权限将被压缩成为匿名使用者，通常他的 UID 与 GID 都会变成 nobody 那个身份（缺省）
all_squash：不管登陆NFS主机的用户是什么都会被重新设定为nobody。
anonuid：将登入NFS主机的用户都设定成指定的user id,此ID必须存在于/etc/passwd中。
anongid：同 anonuid ，但是变成 group ID 就是了！
sync：资料同步写入存储器中。
async：资料会先暂时存放在内存中，不会直接写入硬盘。

客户端NFS服务的测试

- showmout命令对于NFS的操作和查错有很大的帮助，所以我们先来看一下showmount的用法
  showmout
  -a： 这个参数是一般在NFS SERVER上使用，是用来显示已经mount上本机nfs目录的cline机器。
  -e： 显示指定的NFS SERVER上export出来的目录。

客户端NFS服务的测试

- 例如:

```
showmount -e 192.168.0.1
Export list for localhost:
/tmp *
/home/linux *.linux.org
/home/public (everyone)
/home/test 192.168.0.100
```

客户端NFS服务的测试

- mount nfs目录的方法:
  # mount -t nfs hostname(orIP):/directory /mount/point
  具体例子:
  # mkdir /nfs/tmp
  # mount -t nfs 192.168.0.1:/nfs/tmp /mnt/nfs
  # cd /mnt/nfs
  # ls

客户端NFS服务的测试

- 开机时自动连上 NFS:

  如果希望开机的时候，系统就自动挂载 NSF，则需要编辑 /etc/fstab 档。
  例:
  192.168.1.100:/tmp /mnt/nfs nfs defaults 0 0

与NFS有关的一些命令介绍

nfsstat:

查看NFS的运行状态，对于调整NFS的运行有很大帮助

rpcinfo：

查看rpc执行信息，可以用于检测rpc运行情况的工具。

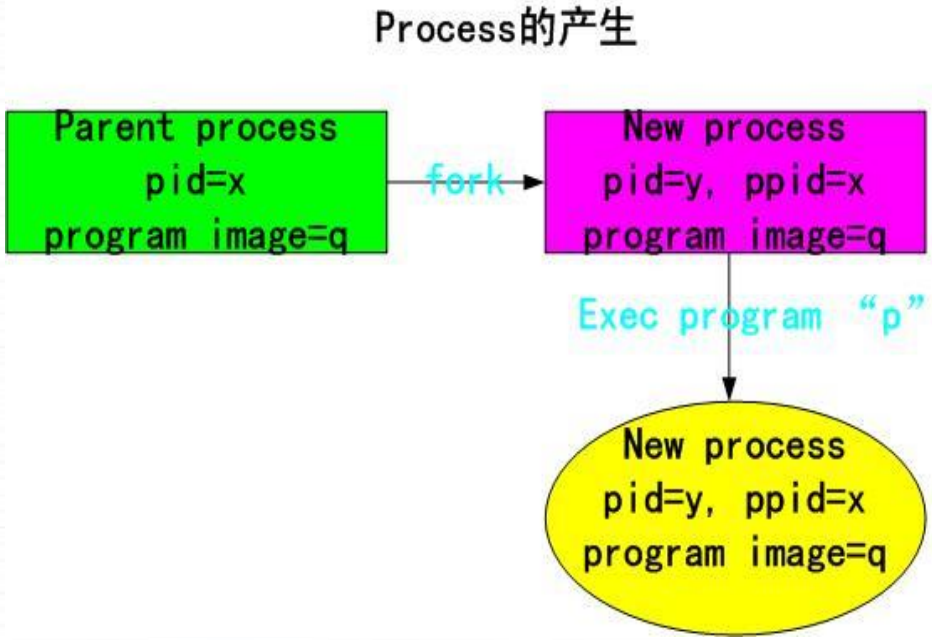
目 录

第六章 系统进程管理

进程的概念

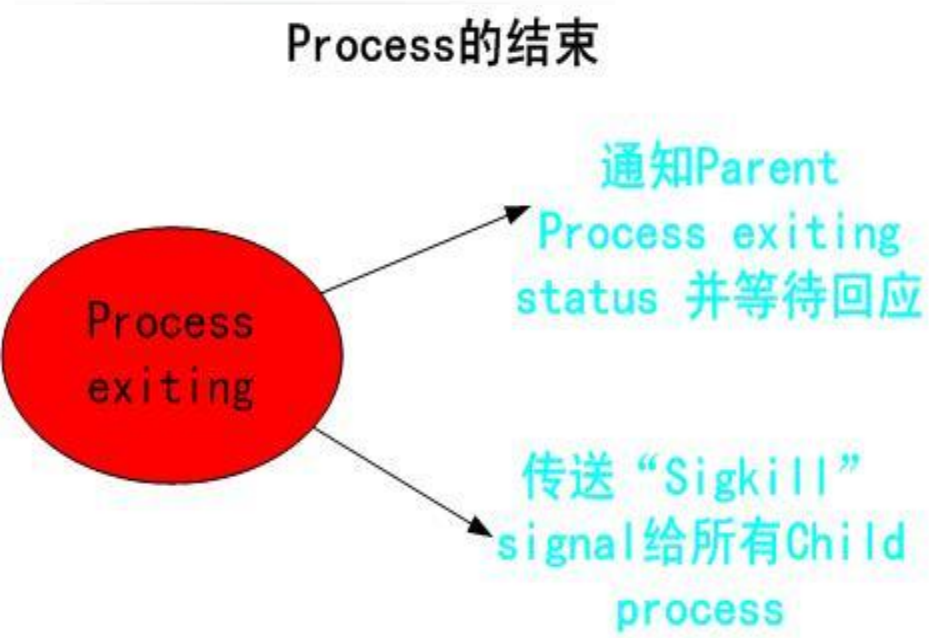
- What is Process
  Linux中Pocess代表一个具有独立记忆空间、可单独运作的”工作空间”(Program)，每一件系统或User的工作均由各种不同的process完成之
- Why Process
  Linux中，很多resource的管理，必须藉由process的控制来完成UNIX每一个程序都有一个唯一的识别代号，我们称之为程序代号(process id or pid)
- How Process Works
  每一process在必要情况之下由其Parent Process产生，完成工作之后会自动释放所有占用的系统资源，结束并系开系统。

进程的产生



- fork-and-exec是Linux中所有process运作的模式。系统bootup的时候，第一个被执行的process(程序)，其pid为1；经由fork-and-exec程序，启动各个必要的process，而形成一个功能完整的操作系统。
- fork是Linux一个系统呼叫(system call)，process fork时，会复制一个跟自己完全一模一样的process (with different pid)，并利用系统呼叫完成之传回值，来区分parent process 与child process，而分别赋予child process不同的功能。
- process于程序执行过程中，利用exec执行另一不同的程序，这个程序并且会完全取代原有程序(with the same pid)。

进程的结束



- process及child process如果因某些原因，parent process在其结束前即已不存在，此 process即成为所谓的Zombie process(or defunct process)，无法正常结束。
- Child process通常会随着parent process结束而结束，因此手动结束process “init”(系统第一个启动的process, pid为1)，将会造成系统当机。(目前大部分的UNIX操作系统会禁止你手动停止init的执行)

进程的监控ps

- [root@node1 ~]# ps aux

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	4752	552	?	S	Apr24	0:01	init [5]
root	2	0.0	0.0	0	0	?	S	Apr24	0:00	[migration/0]
root	3	0.0	0.0	0	0	?	SN	Apr24	0:00	[ksoftirqd/0]

USER: Username of the process owner
PID: Process ID
%CPU: CPU time/elapsed time
%MEM: Fraction of system memory consumed
SIZE: Virtural memory used (K)
RSS: Real memory used
TTY: Terminal port associated with process
STAT: Process state
R Running
S Sleeping
I Idle
Z Zombie
START: Time or date process started
TIME: Total CPU time used
COMMAND: Command line being executed

进程的监控top

```
[root@rac98 ~]# top
top - 17:26:26 up 58 min,  5 users,  load average: 0.00, 0.00, 0.00
Tasks: 120 total,   1 running, 119 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0% us,  0.1% sy,  0.0% ni, 99.9% id,  0.0% wa,  0.0% hi,  0.0% si
Mem:   8170036k total,   291500k used,  7878536k free,    17412k buffers
Swap:  4192956k total,        0k used,  4192956k free,   154424k cached
PID  USER    PR   NI   VIRT   RES   SHR  S  %CPU %MEM    TIME+  COMMAND
5470  oracle    16   0    69660 3260  1912  S     0       0.0       0:00.06     vim
   1     root     16   0     4752  552   460    S     0       0.0       0:00.99     init
```

- q exit "top"
- u *username* Monitor Process Belongs to *username*
- k kill Process specified
- h "top" Online Help
- 1 显示CPU个数

top详解

PID	USER	PR	NI	VIRT	RES	SHR
ID	用户	静态优先级	动态优先级	占用的所有虚拟内存	常驻内存	共享库内存

S ：状态：R运行；S睡眠；D不可中断；Z僵尸；T暂停

%CPU	%MEM	TIME+	COMMAND
占用CPU百分比	占用内存百分比	CPU总时间	执行的程序

top详解

Tasks://任务运行情况
142 total　　//总共142个任务，任务真多，
1 running　　//一个运行
141 sleeping　　//141个睡觉
0 stopped　　//shell下运行一个程序，运行到一半可以按Ctrl+z发送SIGSTOP暂停，然后fg放到前台运行，bg放到后台运行，jobs查看目前运行（暂停）的程序
0 zombie　　//僵尸进程
Cpu(s):　　//cpu使用情况
0.3%us　　//用户进程使用百分比
0.7%sy　　//系统(操作系统)使用百分比
0.0%ni　　//nice，优先极较低的程序使用的CPU
98.7%id　　//空闲进程使用百分比
0.3%wa　　//等待输入输出的CPU时间百分比
0.0%hi　　// 硬件中断
0.0%si　　//软件中断

第七章 系统服务管理

目 录

服务启动方法（1）

- 运行setup——→System services

- 通过空格键来选择或取消择服务

服务启动方法（2）

- chkconfig 命令启动服务

  chkconfig - -level 35 nfs on

  chkconfig - - level 35 nfs off
- 服务开启后，需要将开启的服务在系统中启动，一些服务需要启动/etc/init.d/xinetd，一些服务需要启动自身服务，比如/etc/init.d/nfs restart

rsh服务的管理-简介

- rsh是“remote shell”（远程 shell）的缩写。 该命令在指定的远程主机上启动一个shell并执行用户在rsh命令行中指定的命令。如果用户没有给出要执行的命令，rsh就用rlogin命令使用户登录到远程机上。

rsh 服务的管理–配置

- rsh服务的开启

rsh配置能够满足并行环境的要求，需要开启三个服务，
rsh,rlogin,rexec

[root@node1 ~]#chkconfig rsh on
[root@node1 ~]#chkconfig rlogin on
[root@node1 ~]#chkconfig rexec on
[root@node1 ~]#/etc/init.d/xinetd restart

- 编辑 /root/.rhosts文件

[root@node1 ~]#vi .rhosts
把机器的所有节点的主机名加入
如：node1
node2
…….
nodeN
.rhosts文件权限必须为644

rsh 服务的管理–配置

- 编辑/etc/hosts文件

[root@node1 ~]#vi /etc/hosts

127.0.0.1 localhost

node1 192.168.0.1

node2 192.168.0.2

……

ibnode1 12.12.12.1

ibnode2 12.12.12.2

……

各自网络名字 （各种网络）IP

机群中所有节点的IP和对应的系统名字

- 编辑/etc/hosts.equiv文件

[root@node1 ~]# cat /root/.rhosts > /etc/hosts.equiv

- 编辑/etc/securetty

[root@node1 ~]# vi /etc/pam.d/login(rlogin,rexec,rsh)

注释掉：auth required pam_securetty.so

- 机群中所有节点均需如上方法配置。

ssh服务的管理-简介

- 传统的网络服务程序，如：ftp、rsh和telnet在本质上都是不安全的，因为它们在网络上用明文传送口令和数据，别有用心的人非常容易就可以截获这些口令和数据。而且，这些服务程序的安全验证方式也是有其弱点的，就是很容易受到“中间人”（man-in-the-middle）这种方式的攻击。所谓“中间人”的攻击方式，就是“中间人”冒充真正的服务器接收你传给服务器的数据，然后再冒充你把数据传给真正的服务器。服务器和你之间的数据传送被“中间人”一转手做了手脚之后，就会出现很严重的问题。
- SSH（Secure SHell的缩写）。通过使用SSH，你可以把所有传输的数据进行加密，这样“中间人”这种攻击方式就不可能实现了，而且也能够防止DNS和IP欺骗。还有一个额外的好处就是传输的数据是经过压缩的，所以可以加快传输的速度。SSH有很多功能，虽然许多人把Secure Shell仅当作Telnet的替代物，但你可以使用它来保护你的网络连接的安全。你可以通过本地或远程系统上的Secure Shell转发其他网络通信。

ssh服务的管理-工作机制

- SSH分为两部分：客户端部分和服务端部分。
- 服务端是一个守护进程(demon)，它在后台运行并响应来自客户端的连接请求。服务端一般是sshd进程，提供了对远程连接的处理，一般包括公共密钥认证、密钥交换、对称密钥加密和非安全连接。
- 客户端包含ssh程序以及像scp（远程拷贝）、slogin（远程登陆）、sftp（安全文件传输）等其他的应用程序。
- 他们的工作机制大致是本地的客户端发送一个连接请求到远程的服务端，服务端检查申请的包和IP地址再发送密钥给SSH的客户端，本地再将密钥发回给服务端，自此连接建立。
- 启动SSH服务器后，sshd运行起来并在默认的22端口进行监听，当请求到来的时候SSH守护进程会产生一个子进程，该子进程进行这次的连接处理。

ssh服务的管理-配置

- SSH服务的开启

一般linux系统均自带ssh服务端，仅需将sshd服务开启即可

[root@node1 ~]#chkconfig sshd on

- sshd_config配置文件修改

[root@node1 ~]#vi /etc/ssh/sshd_config

几个重要参数的修改:

PermitRootLogin no ： 限制root用户登录

PasswordAuthentication no ： 仅允许用户以密钥的方式登录

AllowGroups shellusers ： 仅允许同组的登录

AllowUsers username ： 仅允许username用户登录

X11Forwarding yes ： 允许通过ssh调用服务端X11服务

ssh服务的管理-配置

- 通过命令ssh-keygen生成dsa或rsa的密钥对

```
[root@node1 ~]#ssh-keygen  –t dsa
 Generating public/private dsa key pair.
  Enter file in which to save the key (/root/.ssh/id_dsa):
  Enter passphrase (empty for no passphrase):
  Enter same passphrase again:
  Your identification has been saved in /root/.ssh/id_dsa.
  Your public key has been saved in /root/.ssh/id_dsa.pub.
  The key fingerprint is:
  67:e0:be:e5:86:76:76:54:b8:ae:05:5e:1e:79:17:52 root@node1
[root@node1 ~]#cd .ssh/
id_dsa  id_dsa.pub  known_hosts
 [root@node1 ssh]#cat id_dsa.pub >>authorized_keys
[root@node1 ~]#ssh node1 即可不需输入密码登录
```

- 多台服务器之间的密钥对

```
 [root@node1 ssh]#vi config
```

添加如下内容:

```
      CheckHostIP no
      StrictHostKeyChecking no
```

将node1的.ssh目录拷贝到其他节点的/root目录下即可

ssh服务的管理–配置

- 配置ssh密钥对之前，需将该用户目录对其他服务器共享，比如/home目录
- 以test用户为例：

通过命令ssh-keygen生成dsa或rsa的密钥对

```
[root@node1 test]$ssh-keygen  –t dsa
    Generating public/private dsa key pair.
    Enter file in which to save the key (/home/test/.ssh/id_dsa):
    Created directory '/home/test/.ssh'.
    Enter passphrase (empty for no passphrase):
    Enter same passphrase again:
    Your identification has been saved in /home/test/.ssh/id_dsa.
    Your public key has been saved in /home/test/.ssh/id_dsa.pub.
    The key fingerprint is:
    a8:98:1e:2e:74:da:30:32:52:f4:8b:ff:e7:12:ba:98 test@node1
[root@node1 test]$cd .ssh/
id_dsa  id_dsa.pub  known_hosts
[root@node1 test]$cat id_dsa.pub >>authorized_keys
[root@node1 test]$chmod 644 authorized_keys
[root@node1 test]$ssh nodeN 即可在各台之间不许认证登录
```

该用户目录的.ssh/目录中有自动生成的known_hosts文件记录所有节点的密钥

谢谢！！