 linux 相关知识 \*\*\*\*\*\*\*\*\*\*\*

# .基础

eth0 第一块网卡

eth1 第二块网卡

/dev/sda第一块硬盘（SCSI或 SATA硬盘）

/dev/sda1第一块硬盘的第一个分区

如果是hd，说明是IDE硬盘

setup文本模式设置工具(网络、防火墙、服务等)

service network restart 重新启动网络服务(修改网卡等配置信息后，需要重启服务)

网易开源镜像站

http://mirrors.163.com

  CentOS、Ubuntu、FreeBSD等下载

网站扫描

http://news.netcraft.com

可以了解某个服务器用的是什么操作系统

国际超级计算机 TOP500

www.top500.org

可以从中了解到大型服务器使用的是什么操作系统 Linux资料

命令存放目录不一样，决定权限不一样

  root才有权限的命令存放目录:

    /sbin

    /usr/sbin

  所有用户(all users)都可以使用的命令存放目录:

    /bin

    /usr/bin

  其中 bin 是 binary 二进制的缩写

命令提示符

[root@localhost ~]# 超级管理员(root)

[jack@localhost ~]$ 普通用户

linux常见目录

  / 根目录

  /bin 命令保存目录（普通用户可以使用的命令） \*\*

  /boot 启动相关文件(生产服务器最好单独分区，否则如果根分区满了，系统就不能启动了)

  /dev 设备文件保存目录 \*\*

  /etc 配置文件保存目录 \*\*

  /home 普通用户的家目录(宿主目录) \*\*

  /lib 系统库保存目录

  /mnt 系统挂载目录(挂载U盘、光盘等) \*\*

  /media 也是挂载目录

  /root 超级用户的家目录 \*\*

  /tmp 临时目录

  /sbin 命令保存目录（超级用户才能使用的命令存放目录）

  /proc 一个虚拟文件系统, 存在于内存之中而不是硬盘上(内核相关数据结构)

  /sys 虚拟的基于内存的文件系统，它的作用与 /proc 有些类似，

但除了与 proc 相同的具有查看和设定内核参数功能之外，

还有为 Linux 统一设备模型作为管理之用

  /usr 系统软件资源目录(我们自己安装的软件，通常也装在这里)

    /usr/bin/ 系统命令（普通用户）

    /usr/sbin/ 系统命令（超级用户）

  /var 系统相关文档内容

    /var/log/ 系统日志位置

    /var/spool/mail/ 系统默认邮箱位置(在里面每个用户都有一个邮件目录)

Linux特点：\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

内核只认识数字(例：文件的i 节点)

修改和设置需要写配置文件，改后通常需要重启相关服务

在Linux中，所有的东西，都是文件

  /dev这个目录包含了所有Linux系统中使用的外部设备

  设备文件分为两种：块设备文件(b)和字符设备文件(c)

  /dev/sd[a-z]：SCSI或SATA设备

  /dev/hd[a-t]：IDE设备

  /dev/null：无限数据接收设备,相当于黑洞

    find / -name access\_log 2>/dev/null

    find 命令在/目录下查找名为 access\_log 的文件，并且错误输出不会显示出来（文件描述符 2 被重定向到/dev/null）

  /dev/zero：提供无限零资源

    dd if=/dev/zero of=test.txt bs=1k count=2

每块1k，读2 块输出到test.txt文件中, 产生 2KB 大小的文件 text.txt

tab键自动补全命令或文件名

\*\*\*\*\*\*\*\*\*\*\* 常用命令 \*\*\*\*\*\*\*\*\*\*\*\*\*

ifconfig 查看网络信息

ctrl+c 终止正在执行的命令

  例： ping 127.0.0.1 执行后，不会停，按 ctrl+c终止

==============================

# 文件处理命令

==============================

ls 查看目录或文件信息 list

  -a 显示所有文件all

  -l 显示详细信息long

  -d 查看目录信息directory 而不是查看该目录下有哪些文件

  -h 文件大小单位使用KB MB 等人性化显示 (默认单位:字节)

  文件类型(第一个字符)

    d 目录

    - 二进制文件

    l 软链接文件link

  文件权限

    r read可读

    w write可写

    x execute 可执行

  用户分类

    u user(owner)所有者

    g group所属组

    o others其他人

  分页查看目录内的信息：ls -l /etc|more 分页查看/etc目录

例ls -l /etc/yum.conf 得到如下结果

-rw-r--r-- 1 root root 346 2010-04-05 yum.conf

  第 1部份是文件类型和权限

  第 2部份是文件硬链接数

  第 3部份是文件所有者 Linux资料

  第 4部份是文件所属组

  第 5部份是文件大小

  第 6部份是文件创建或最后修改时间 提示ctime是change time不是 create time

  第 7部份是文件名

cd 改变目录 change directory

  cd ~ 回家(用户的宿主目录) 或直接cd

  cd – 上次目录

  cd .. 上级目录

pwd 显示当前目录 print working directory

mkdir 创建目录make directories

  mkdir -p 11/22/33/44 加-p 是递归建立目录

touch 创建文件

cp 复制 copy

  复制目录需要加 -r 如：cp -r /etc /test/

  复制目录下的所有文件 cp –r /etc/. /test/

  cp -p file file2 让新文件时间都一样

 -r 复制目录 或 -R

    -p 连带文件属性一起复制

    -d 若源文件是链接文件，则复制链接属性

    -a 相当于 -pdr

  cp aa /test/ 原名复制

  cp aa /test/bb 改名复制

  如果是做备份 建议目标文件名加上时间：

YYYYMMDDXX 四位年两位月两位日两位当天修改次数

mv 移动 move (剪切或改名)

操作目录不需要加-r 选项，注意与 cp区别

可以在移动的同时改名

例： mv oldname newname

rm 删除文件remove

  rm -f 不确认直接删除

  rm -rf 删除目录（不确认，直接删除）

  rm -rf \* 删除当前目录下的所有文件或目录 (千万不要在根目录下执行这个操作)

rm -rf /test/\* 删除/test下的所有文件或目录

cat 查看文件所有内容，此命令不能完整的显示太长的文件

  cat /etc/services

more 分页查看

  下一页：空格或 f

  下一行：回车

  向上翻：b

  退出:  例：more /etc/services q 或Q

head 查看前几行

  例 head -20 /etc/services

tail 查看最后几行

  例 tail -30 /etc/services

  动态显示:f

  例：tail -f my.log 监视日志文件，当文件被添加新行，将刷新显示新内容

测试时，可以用另一个用户登录执行命令echo `date`>>my.log，tail会刷新

ln link

  创建软链： ln -s 源文件 新文件

  创建硬链接：ln 源文件 新文件

软链接：

  文件类型是 l 用ls –l能看到

  1.软链接文件权限都是 rwxrwxrwx ，实际访问，取决于源文件的文件权限

  2.文件名后，有一个 箭头 ( ->) 指向源文件

  3.文件很小，只是一个符号链接 Linux资料

  4.时间是创建软链接的时间

  类似于 Windows的快捷方式

硬链接：属于硬链接的两个或多个文件，权限与属性都一样，硬链接数也一样

  类似于拷贝(cp -p)加上同步更新(操作源文件或新文件，结果都是一样的，两个文件都会变)

  删除其中任何一个，不影响另一个。软连接如果删除源文件，则软连接不能访问了

  为什么可以同步更新 用ls -i 查看inode(i节点)是一样的，是一个数字表识,

  linux 内核只认数字，内核要处理的任何对象，都需要有一个数字标志,用户 uid 组 gid 进程 pid

  linux 中，每个文件，必须有一个 i节点，除非文件系统出错，如果没有了 i节点，文件是不能访问的

  一个 i节点，可以对应多个文件，硬链接文件i节点一样的，所以可以同步更新

  同一个 i节点，被映射到了多个文件，内核一看，i节点都一样，所以全都一起更新

硬链接 -> 增加了文件的i节点映射

删除硬链接 -> 减少了一个映射

硬链接，只能链接二进制文件，不能链接目录

软链接可以跨文件系统(不同分区)，硬链接不能。假如 windows 中也有硬链接，则 C 盘的文件不能生成一个硬链接

到D盘

=================================

# Linux 用户及用户组

=================================

 用户信息文件：/etc/passwd

 用户密码文件：/etc/shadow

 用户组文件：/etc/group

/etc/passwd文件格式

字 段 含 义

用户名 用户名，字母和数字组成

密码 密码位(只是标识位，不存放密码)

UID 用户标识

GID 缺省组标识

描述 描述信息，可存放用户全名等信息

宿主目录 用户登录系统后的缺省目录

命令解释器 用户使用的Shell，默认为bash 用户类型

超级用户（root，UID=0）

 普通用户（UID 500-60000）

 伪用户（UID 1-499） 伪用户

1、伪用户与系统应用和程序服务相关

如：bin、daemon、shutdown、halt等，与Linux

系统应用相关

如：apache、ftp、mysql及sshd等，与Linux 系

统服务相关

2、伪用户通常不需要或无法登录系统

3、可以没有宿主目录

/etc/shadow文件格式

字 段 含 义

用户名 用户名

密码 加密密码

最后一次修改时间 用户最后一次修改密码的天数

最小时间间隔 两次修改密码之间的最小天数

最大时间间隔 密码保持有效的最多天数

警告时间 从系统开始警告到密码失效的天数

帐号闲置时间 帐号闲置时间

失效时间 密码失效的绝对天数

标志 一般不使用

/etc/group文件格式

字 段 含 义

组名 用户组名称，建议按功能或

组织划分和命名

组密码位 组密码一般不使用

GID 用户组标识

组内用户列表 属于该组的所有用户列表，

多个用逗号分隔 用户组管理命令

添加用户组 groupadd

 groupadd webadmin

    创建用户组webadmin

   cat /etc/group

删除用户组： groupdel 组名

 添加用户

useradd 设置选项 用户名 useradd G www jim

g：缺省所属组名

G：指定用户所属多个组

c：描述信息

e：指定用户失效时间 YYYY-MM-DD

useradd -g webadmin jack

useradd -g webadmin -G group1,group2 user1

passwd 用户名

修改用户信息

 usermod

usermod -g webadmin jack 改变默认组

usermod -G softgroup,adm jack 修改从组

用户jack将在三个组中，不管以前属于多少个组

usermod --help

man usermod

-l 改名 -c 改备注 -e 过期时间 -L 锁定 -U 解锁

-aG 追加到组中 用户组管理命令

    gpasswd (UNIX中没有此命令)

     -a 添加用户到组中

    gpasswd -a user group

     -d 从组中删除用户

    gpasswd -d user group

 删除用户

userdel -r 用户名

-r：删除用户宿主目录

 用户管理命令

 id 查看用户UID、缺省组、所属组

 su 切换用户（su - 环境变量切换）

 newgrp 用户属于多个组时，切换默认组

==================================

# 权限管理

==================================

chmod 修改权限change mode

  用户类型分为 ugo三种 user group other 所有者 所属组 其他人

  权限分为 rwx三种 read write execute 可读 可写 可执行

  操作有三种：添加(+) 去除(-) 覆盖(=)

  组合以上三种，可以灵活设置文件权限

  如：

  给所属组添加写权限: chmod g+w filename

  将其它人读权限去掉: chmod o-r filename

  将所有者权限设 rwx: chmod u=rwx filename 不管以前是什么权限了，通通改为现在这样

  将全部人的权限设为 rwx : chmod a=rwx filename a all表示所有人

  重点是通过数字的方法来授权

  r 4 2 的2次方

  w 2 2 的1次方

  x 1 2 的0次方

  将每一个用户的三种权限，用一个数字表示

  如 rwx --> 7 4+2+1

  文件权限例如:

  rwxr-xr-x 755 默认创建新目录的权限

  rw-r--r-- 644 默认创建新文件的权限

  要求掌握：给出字母，写出数字表示 给出数字，能用字母表示。 Linux资料

递归修改文件夹内所有文件权限：

   Chmod –R 777 mytest

  实验：

  root用户创建一个目录(默认rwxr-xr-x)，在目录中新建一个文件，然后将新建的文件设为777，普通用户能否删

除该文件?

  实验需要相关知识:

  useradd 用户名 #添加用户

passwd 用户名 #为用户设置密码

  su - 用户名 可以切换到其它用户身份 (管理员切换到普通用户不需要密码 exit 回到root身份)

  如果直接 su 用户名 中间没有" - " ，也可以切换用户，但环境变量没有切换，还是原帐户的环境变量

查看环境变量： echo $PATH

  答案：不能删除

  实验2:

  root用户创建的目录，将目录改为777，将文件权限为644(rw-r--r--)，切换到普通用户，是否可以删除该文件？

  答案：可以删除

  因为 rwx权限对于目录和文件，实际意义是不一样的

  文件:

  r cat more head tail 查看文件内容

  w echo vi 修改文件内容

  x sh (execute) 执行

  目录:

  r ls find tree 列出文件

  w touch mkdir rm 创建删除文件

  x cd 进入目录

  对文件有写权限，只代表可以修改这个文件，要想删除文件，需要对文件所在目录有写权限

  只要对目录有写权限，目录中的文件，就算是 root创建的，普通用户一样能删除

  所以，目录一般都同时有rx权限，如果一个目录只有 r却没有 x，你能用ls查看，却不能 cd 进去！(用普通用

户测试)

  [jack@localhost test]$ cd a

-bash: cd: a: 权限不够

[jack@localhost test]$ ls a

123

  对目录 w权限，就意味着需要有 rwx

对脚本执行，需要rx 并对该目录有rx权限

chown 改变所有者 change owner

  系统中自带一个特殊用户nobody

  useradd 用户名 添加用户

  passwd 用户名 设置密码

  例: Linux资料

  chown nobody filename 把filename 文件所有者改为 nobody用户

chgrp 改为变所属组 change group

  chgrp adm filename 把filename文件所有组改为 adm组 (adm是系统内置的一个组)

chmod chown chgrp 都可以加-R 递归修改

缺省创建文件的权限：

umask 显示0022

umask -S 显示u=rwx,g=rx,o=rx

    有些linux并没有-S 选项

0022

  第一个 0 特殊权限位(SetUID 、SetGID 、粘着位) 在用户管理中讲解

  后台三位(022)，是掩码值，表示用户权限位，是权限掩码值

   777

  -022

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

    755 就是创建文件的默认权限

我们创建一个目录 mkdir test 默认是755 权限

但新建的文件 touch file 默认是 644 权限

linux权限规则：

缺省创建的文件不能授予可执行权限

所以是 755减去 x权限 755-111=644

这是一个安全机制

一般不改 umask默认值

不建议改变这个 755

想做试验改默认的为 750

   777

  -750

     --------

    027

执行umask 027，缺省创建的文件权限就改了。做完实验记得改回来 022

======================================

# 搜索命令

======================================

which 定位到命令的绝对路径 [wɪtʃ] adj. 哪一个；哪一些

  linux 中，所有东西都是文件，命令也是文件

  可以根据文件路径判断执行权限，权限分为2 种，一种是所有用户都可以用，一种是只有管理员root才可以用。

当然可以通过前面学习的权限改变。

  例如 which ls

    alias ls='ls --color=tty'

         /bin/ls

whereis 与which类似

  如 whereis ls

    ls: /bin/ls /usr/share/man/man1/ls.1.gz /usr/share/man/man1p/ls.1p.gz

两者区别:

  which除了找到该命令，还能找到与之相关的别名记录

  whereis 还能找到帮助信息文件（man里面的就是帮助文件）

find

  普通文件或命令都可以找到

  需要指定搜索路径和条件

  尽量不要在根目录/中查找，查找得慢，会占用太多服务器资源

  如果需要在服务器中全盘查找，可以写一条计划任务在凌晨执行(计划任务后面讲)

  范围越小越好

匹配条件，越精准越好

  -name 根据文件名查找 \*任意字符 ? 单个字符

  例 find /etc -name init 这是精确查找

      find /etc -name init\* 查找以 init开头的(\*代表0个或多个任意字符)

      find /etc -name init??? 用?号代表任意一个字符

  -size 根据大小查找 单位是 block 数据块 一块是512 字节 10M=1024k\*10\*2=20480 block

  1M -> 1024k -> 2048 块 (1块是0.5k 也就是512 字节)

100M -> 102400k -> 204800块

  例 find /etc -size -20480 查找小于10M的文件

      find /etc -size +204800 查找大于100M的文件

    注意+-号，如果没有，是精确这么大，通常都会带上+或-号，表示一个范围

  -user 根据所有者查找

  例 find /home -user zhangsan

  按时间查找

     按天 ctime atime mtime

     按分钟 cmin amin mmin

  c change 改变 表示属性被修改过：所有者、所属组、权限

  a access 访问 被访问过(被查看过)

  m modify 修改 表示内容被修改过

  例：find /etc -mmin -120 查找120 分钟内被修改过的

       find /etc -mtime +7 超过7 天没有改过的文件

  -type 根据文件类型查找

  例: -type f 二进制文件

    -type l 软链接文件

    -type d 目录

  连接符（逻辑连接符: -a -o 命令执行连接符: -exec -ok）

  逻辑连接符:

  -a (and 逻辑与 ) -o (or 逻辑或)

  例： find /etc -size +163840 -a -size -204800 在/etc下查找大于80M小于100M的文件

  对查找到的文件，进一步处理

  find ... -exec 命令 {} \;

      {}表示 find查询的结果集

      \是转义符 一个命令或符号使用其本身的意义，不使用别名

      ;分号是表示语句的结束

      是固定格式，只能这样写。注意中间的空格。

  转义符的作用是什么？

  在 linux中有一个别名机制，如rm删除文件，执行的却是rm -i(用which rm 可以查看命令别名)，使用 rm删除

文件前会提示，就是因为 rm -i这个参数。如果想使用命令原意，可以在加\转义，

  如：\rm test.txt 则不会提示，直接删除

  找到文件后查看详情:

  find /etc -name inittab -exec ls -l {} \;

  找到 testfile 后，直接删除

  find /test -name testfile -exec rm {} \;

  找 samlee 用户的所有文件并删除

  find /home -user samlee -exec rm -rf {} \;

  找 samlee 用户的所有文件并删除，删除前会一个提示确认

  find /home -user samlee -ok rm -rf {} \;

  结合使用:查二进制文件并显示详情

  find /etc -name init\* -a -type f -exec ls -l {} \;

  根据 i节点查找

  如果一些特殊文件

    touch "a b"

    touch -- -abc

  无法直接删除

    rm a b

    rm -abc

  可以这样删除

    rm "a b"

    rm -- -abc

  如果你不知道是怎样创建的，则可以先查看 i节点

    ls -i 查到结果例如 2310630 a b

    find ./ -inum 2310630 -exec rm {} \;

locate 这是linux特有的命令 [ləuˈkeit] 确定…的位置

  locate filename

  速度比 find快，但新创建的文件，是找不到的，需要更新数据库

  updatedb 更新 locate 的数据库 系统有计划任务，会定期执行updatedb

grep ftp /etc/services

  查看 services中查找ftp相关的行

  后面有大量应用(特别是在管道应用中)

  grep php /root/install.log 在日志文件中查找 php字符

-i 不区分大小写查找

grep -v "^#" /etc/httpd/conf/httpd.conf 不显示行首为#号的 (反向匹配)

  | 管道符

    将前一个命令的结果，作为后面命令的输入

  ls | grep aa

  netstat -tlun | grep 80

补充命令：

    netstat 查看网络状态的命令

      -t 查看tcp端口

      -u 查看udp端口

      -l 监听

      -n 以 IP和端口号显示，不用域名和服务名显示

======================================

# 帮助命令

======================================

man manual 缩写 [ˈmænjuəl] n. 手册；指南；

  可以获取命令的帮助，也可以获取配置文件的帮助

  man ls 查看ls命令的帮助

  man services 查看配置文件service 配置文件的的帮助 配置文件名前面不加路径

  会调用 more 命令来显示帮助文档

    下一页：空格或f

    下一行：回车

    退出:q 或Q

NAME 命令的作用

  DESCRIPTION 查看命令选项作用 /选项 就可以找到 需要查看的选项

  main passwd

    是查看passwd命令的帮助，还是 /etc/passwd文件的帮助？

  优先查看的是命令的帮助

  帮助分多种:

    1 是命令的帮助 默认就是 1

    5 配置文件的帮助

  如 man 5 passwd 就是看配置文件的帮助了

info 与 man用法类似，linux才提供此命令

whatis

  例：whatis ls 显示ls命令的简单描述信息

命令 --help 显示某个命令的选项

  如 ls --help

apropos fstab

查看fstab相关的文件信息

makewhatis

  建立whatis 和 apropos搜索使用的数据库，当使用这两个命令发生错误时，就是whatis database 没有建立

  新安装的软件，帮助信息没有更新时，就需要调用这个来更新

shell命令解释器

如何判断是否内置命令 which whereis 没有路径 no cd in 说明是 shell内置命令

如：cd umask

help 查看 shell内置命令的帮助

  shell是什么？

  最底层是硬件，上一层是 linux 内核，再上一层是shell(命令行的交互)

  shell也是一个软件，有很多种，不同的shell，提供的命令也不完全一样

  有些命令不在任何目录中，而是安装 shell自带的

  这种命令在获到帮助时，有区别，不能用 man查看，如 man cd 将无法得到cd的帮助信息

  应该用 help cd

# 压缩解压

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

linux下所有的压缩格式，WinRAR 都支持

gzip

  .gz 格式

  压缩文件:

  gzip 文件名

  特点：

  1.只能压缩文件，不能压缩目录

  2.不保留原文件

  3.不用指定生成的新文件名

  解压：

  gunzip 文件名 或 gzip -d 文件名

tar

  把目录打包成一个文件，后缀名是 .tar

  打包后，可以再压缩，就是 .tar.gz

  网上的源代码包，经常就是这个格式

  -c 创建打包文件 必须项 create

  -v 显示详细信息

  -f 指定压缩后的文件名 必选项

  -z 打包同时压缩

  例：

  tar -czvf 目标文件.tar.gz 目录名

  如果 Unix服务器不支持打包的同时压缩，就只能分两步了，也可以用后面学的命令连接符处理

  解压

  tar -xzvf 压缩文件名 就是把 c 换成了x

  本地简单目录备份，推荐用 tar

zip

  windows与linux默认都支持

  可以压文件，也可以压目录，与 gzip不同，zip会保存源文件

  压目录加一个 -r选项

  例：zip services.zip /etc/services 先写生成的目标文件

  压缩目录

  zip -r test.zip /test 压缩/test目录

  解压:

  unzip 文件名

bzip2

  是 gzip的一个升级版，加一个-k选项后，可以保存源文件，使用方法与gzip一样

  压缩比很高，网上一些比较大的安装包，很多都是.bz2格式

  也 gzip一样，也不能压缩目录，想压缩目录，先用tar 打包

  例:bzip2 file.bz2 file 或bzip2 -k file.bz2 file

  解压:

  bunzip2 file.bz2 或 bunzip2 -k file.bz2

  如果你生成的目标文件，没有写扩展名，你无法用眼睛识别是什么文件了

file

  例：file 文件名 可以判断这是什么文件

# 网络通信指令

write

  给用户发信息

  例 write user1 回车后，输入内容 CTRL+d 结束

  没有登录的用户，是不会成功的。是实时的通讯工具。

who 当前登录系统的用户列表

  tty 本地终端 pts 远程登录

wall ( write all)

  写给所有人，发广播

  例：wall 内容

  在线用户都会收到

ping

  ping ip地址 ctrl+c 结束

  在 window 中，默认4 次就结束了，linux下，会一直ping

  不是能 ping通，就说明能通了，要查看延迟时间和丢包率

  ping不通，也可能是防火墙或其它原因

  如果 ping其它电脑不通，先检查自己，ping自己的ip，自己 ip通，说明本机ip协议没问题

  -c 指定ping 几次 count

  如 ping -c 3 192.168.1.1

  -s 指定发包大小

  如 ping -s 65507 192.168.1.1

ifconfig

  window中 ipconfig /all

  直接 ifconfig或 ifconfig eth0

  临时改 ip

  ifconfig eth0 192.168.1.8

  ifconfig eth0 192.168.1.8/24

  ifconfig eth0 192.168.140.252 netmask 255.255.255.0

关机

shutdown -h now

init 0

  服务器别直接关机,先通知一下其它用户

重启

reboot

==================================================

# 挂载命令

  linux 所有存储设备都必须挂载使用，包括硬盘

  命令名称：mount

  命令所在路径：/bin/mount

  执行权限：所有用户

  光盘挂载

  /dev/sda1 第一个scsi硬盘的第一分区

  /dev/cdrom 光盘

  /dev/hdc 光盘

  mount -t 文件系统 设备描述文件 挂载点（已经存在空目录）

  mount -t iso9660 /dev/cdrom /mnt/cdrom

  很多时候，不用指定文件系统 mount /dev/cdrom /mnt/cdrom

  光盘卸载

  umount /dev/cdrom

  umount /mnt/cdrom 强调：退出挂载目录，才能卸载

识别 USB存储设备

包括 USB硬盘、U盘、MP3 播放器等

/dev/sda /dev/sdb ……

sd: SCSI SATA hd: IDE

a: 第一块硬盘

1： 第一个分区

# fdisk -l

使用 mount命令挂载U盘

# mount -t vfat /dev/sdb1 /mnt/usb

很多时候，不用指定文件系统 mount /dev/sdb1  /mnt/usb

通过挂载点目录访问 U盘的内容

# ls /mnt/usb

使用 umount命令卸载U盘

# umount /mnt/usb

======================================================

# shell技巧

有很多种 shell

cat /etc/shells 列出当前安装的shell有哪些

几乎所有 linux缺省都是bash (Bourne-again Shell) ，非常适合于一些管理操作

可以按 Ctrl+D注销 exit

一些比较老的 unix，缺省都是sh

bash提供如下功能

命令补全

  Tab键

  命令补全、文件名补全

  如果不是唯一的，按两下 Tab，全列出来

清屏

  clear 命令 快捷键 ctrl+l

  windows中是 cls

清除一行

  ctrl + u 命令敲错了，不用一个一个字符删除了

CTRL+C 终止命令

CTRL+A 光标到行首

CTRL+E 光标到行尾

CTRL+K 剪切光标后内容

CTRL+Y 粘贴剪切的内容

命令历史

  history 列出所有的命令

  !序号 执行历史中第几个命令

  按向上(或向下)箭头，翻出历史记录

命令别名

  alias 列出系统中的别名

  ls为什么可以显示颜色？因为是 ls --color=tty的别名

  可以看到 ll 是 ls -l --color=tty 的别名

  添加一个别名:

  例 alias copy=cp 就可以用 copy 来复制文件了

  如果有参数：

  alias drm="rm -rf"

  删除别名:

  unalias copy

别名信息，保存在用户宿主目录 ~/.bashrc 文件中

全局别名，保存在/etc/bashrc 文件中

输入输出重定向

  Shell对于每个进程预先定义了3 个文件描述字

  0 标准输入 STDIN 键盘

  1 标准输出 STDOUT 显示器

  2 标准错误输出 STDERR

  重定向，就是改变这个标准设备，输入不从键盘，输出不显示到显示器上

  输出重定向

    > 或 >>

  例： echo hello 直接显示到显示器

    echo hello>test.txt

  ls -l /tmp>files.txt

  more files.txt 查看文件内容

  把/website 下大于100M的文件列表

find /website -size +204800 >/backup/100M+.file.list

  如果不想替换以前的内容，是追加，用 >>

  输出重定向用得挺多的，一个人管理几十台服务器，定时将系统状态保存到文件中，然后发邮件经管理员

date 查看系统时间

设置系统语言

LANG= 后面啥也不写，表示使用系统默认值，系统默认是英文

  输入重定向 (不从键盘输入内容了)

  例：wall < /test/msg 从一个文件读取内容发广播

  这样，就可以用计划任务，把一年的节日祝福语写上，到时自动发

  错误输出重定向

  2> 或 2>>

  例：自动备份 通常用计划任务在凌晨自动执行

  cp -R /usr /backup/use 2>>/bak.error

  将错误信息，定向到一个文件中

  ls /aaaaaa 2>ls.err 如果/aaaaa这个目录不存在，则将错误信息保存到ls.err中

  这里面的2，就是前面讲的 Shell对于每个进程预先定义了3个文件描述字

  0 和1都可以省略，2 不能省略了

管道 |

  将一个命令的输出，传送给另一个命令，作为另一个命令的输入，可以连接多个命令

  用法:

    命令1|命令2|命令3...

  例:

  ls -l /etc | more 文件太多，一次看不完，用 more来查看 下一页：空格或f 下一行：回车 退出:q 或 Q

  ls -l /etc | grep init 只显示init相关的行

  ls -l /etc | grep init | wc -l 查看init相关的有多少行，也就是包含init的文件有多少个

  who | grep root 只显示root的登录信息

wc -l 统计文件有多少行

who | grep root | wc -l 查看root 用户登录了几次

命令连接符

  ; 分号 多个命令依次执行

    例: pwd ; ls ; date

  && 前面执行成功，才执行后面的命令，如果第一个失败，则不执行第二个

    write user1 < /home/jack/love.txt && rm /home/jack/love.txt 情书发成功，就删除

    ls && pwd 第一个成功后，将执行 pwd命令

  || 前面执行失败，才执行后面的命令

    write mary < /home/jack/love.txt || mail mary< /home/jack/love.txt 情书发失败，则发邮件

  () 改变命令的执行顺序

  管道的优先级，高于连接符

命令替换符

  将一个命令的输出作为另一个命令的参数

  命令1 `命令2`

  命令1 的参数，是命令2的执行结果

  例： ls -l `which touch` 找到 touch命令的路径，然后用 ls查看属性

  看起来有点类似管道，不过顺序相反，会先执行后面的

Sudo授权

配置文件：/etc/sudoers，编辑配置文件命令visudo，

普通用户使用命令sudo

格式：

用户名 主机地址=命令（绝对路径,用which 查看）

jack ALL= /bin/vi /etc/httpd/conf/httpd.conf

通过sudo授权用户可以修改配置文件和管理服务脚本：

visudo

helen localhost=/bin/vi /etc/httpd/conf/httpd.conf, /etc/rc.d/init.d/httpd start , /etc/rc.d/init.d/httpd reload , /etc/rc.d/init.d/httpd configtest , /etc/rc.d/init.d/httpd status

fay localhost=/etc/init.d/httpd start,/etc/init.d/httpd reload

以上为授权用户helen可以修改apache配置文件，可以启动apache、修改apache配置文件后加载生效、以及检测apache配置文件语法错误和查看apache状态。

 Sudo示例

不要将vi授权给普通用户，否则可以修改任何配置文件

   sudo vi /etc/passwd

   sudo vi /etc/shadow

如果想让普通修改某个配置件文件，写上具体参数

  /bin/vi /etc/httpd/conf/httpd.conf

赋予user1改普通用户的密码，但不允许修改root的密码

user1 ALL=/usr/bin/passwd [A-Za-z]\*, !/usr/bin/passwd "", !/usr/bin/passwd root

 ACL权限设置

1. 设置文件的acl权限

     setfacl -m <rules> <files> (modify)

  setfacl -m user:用户名:权限 文件名

  setfacl -m g:组名:权限 文件名

  例 setfacl -m user:jack:rx /backup

2. 查看文件的acl权限 (权限最后有一个+)

     getfacl <files> 设定ACL使用命令setfacl

 -m：设定权限

 -x：移除权限

setfacl -x user:jack /backup

 -b：移除所有权限

-R：递归设置(对于目录)

setfacl -m d:u:jack:rwx -R /test

继承，对新创建的文件或目录也有效

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

# John密码破解

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

http://www.openwall.com/john/

tar -zxvf john-1.7.9.tar.gz

cd john-1.7.9/doc

more INSTALL #查看安装说明，这不是一个标准的源代码安装包

cd ../src

make | more

make linux-x86-sse2 #选择对应操作系统平台

cd ../run

useradd test #添加一个用户

passwd test #设一个简单的密码 如rain

#将密码信息提取出来

grep test /etc/passwd > test.passwd

grep test /etc/shadow > test.shadow

#将shadow中的密码转换出来

./unshadow test.passwd test.shadow >test.john

cat test.john

./john test.john #开始破解

=====================================

# 服务与进程管理命令

=====================================

Linux 服务与进程

 查看用户信息 w

w显示信息的含义

load average：分别显示系统在过去1、5、15

  分钟内的平均负载程度。

up：系统连续运行时间。

 查看用户信息 w

 w显示信息的含义

IDLE：用户闲置的时间。

JCPU：以终端代号来区分，该终端所有相 关的进程执行时，所消耗的CPU时间

PCPU：CPU执行当前程序耗费的时间

WHAT：用户正在执行的操作

 查看用户信息：w 用户名

查看进程信息 ps

常用选项

a：显示所有用户的进程

u：显示用户名和启动时间

x：显示没有控制终端的进程

e：显示所有进程，包括没有控制终端的进程

l： 显示详细信息 查看进程信息 ps

ps应用实例

# ps -l 查看隶属于自己进程详细信息

# ps -le 或 -aux

查看所有用户进程的详细信息

# ps -le | grep 进程名

查看指定进程信息

# ps uU 用户名

查看系统中指定用户执行的进程

 查看进程信息 ps

ps常用输出信息的含义

PID：进程号

PPID：父进程进程号

TTY：进程启动的终端

STAT：进程当前状态

   S休眠状态，D不可中断的休眠状态，R运行

状态，Z僵尸进程，T停止

NI：进程优先级 查看进程信息 ps

ps常用输出信息的含义

TIME：进程自启动以来占用总的CPU时间

COMMAND/CMD：进程名

USER/UID：用户

%CPU：占用CPU时间和总时间的百分比

%MEM：占用内存与系统内存总量的百分比 kill – 关闭进程

关闭进程：kill PID

 kill -9 PID（强行关闭）

 kill -1 PID（重启进程）

结束所有进程：killall 进程名

 查找服务进程号：pgrep 进程名

关闭进程：pkill 进程名

 进程的挂起和恢复

进程的挂起（暂停）和终止

挂起（Ctrl+Z）

终止（Ctrl+C）

后台执行：命令 &

查看挂起及后台执行进程（jobs）

进程的恢复

恢复到前台继续运行（fg）

恢复到后台继续运行（bg）

top

   作用：进程状态显示和进程控制，默认每3

秒钟自动刷新一次（动态显示）

u：查看指定用户的进程

k：终止执行中的进程

d：改变刷新的时间间隔

h：获得帮助

q：退出

 服务管理

 服务分类：

    系统默认安装的服务

      独立的服务

      基于xinetd的服务

    源代码包安装的服务 如何确定默认安装服务类型

#chkconfig --list

列出在每一种运行级别下服务是启动还是关闭

chkconfig --list | grep atd 查看计划任务

 独立的服务的启动

1.#/etc/rc.d/init.d/httpd

start|stop|status|restart

2.#service httpd start|stop|status|restart 基于Xinetd服务的启动

＃vi /etc/xinetd.d/telnet

  将 disable = yes

  改为 disable = no

#service xinetd restart 服务的自启动

独立的服务自启动

1）#vi /etc/rc.local

/etc/rc.d/init.d/httpd start

2）#chkconfig --level 2345 httpd

on|off 服务的自启动

基于Xinetd的服务的自启动

 #chkconfig telnet on|off 源代码安装的服务

使用源代码包安装方式的服务的启动

例：

  #/usr/local/apache/bin/apachectl

start|stop|restart|status

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

# 计划任务

为什么要设置计划任务

  定时备份，定时重启

  管理任务，定时将服务器信息发邮件，每天只需要收邮件，就能了解很多台服务器的信息

  ...

主要分两种

  一次性计划任务

    at

  周期性计划任务

at

  指定的时间执行一次

  指定时间的方式

    绝对计时方法:

    midnight午夜 noon中午 teatime 下午茶(16:00)

    hh:mm [today]

    hh:mm tomorrow

    hh:mm MM/DD/YY

    相对计时方法:

    now + n minutes

    now + n hours

    now + n days

  具体看帮助 man at

  系统时钟 date

  设置日期 date -s '2013-05-14 11:27:00'

  硬件时钟 clock

  clock --systohc 将系统时间存回硬件时钟内

  clock --hctosys 系统时间设成和硬件时钟一致

  指定在今天下午 17:30 执行某命令，假设当前时间是2013 年1月 15日

  at 17:30

  at 5:30pm

  at 17:30 today

  at now + 3 hours

  at now + 180 minutes

  at 17:30 01/15/13

  at 17:30 15.01.13 如果用(.)表示日月年

  at now + 5 minutes回车后，输入要执行命令，写完按 ctrl+d保存

  命令最好写绝对路径(例如which ls将查询到ls命令在哪里)

  例 at now + 1 minutes

  /usr/bin/wall hi,this is a test. 一分种后发一个广播

  查看计划任务进程是否开启 ps -le | grep atd

  启动进程 /etc/rc.d/init.d/atd start

  默认都会安装计划任务并启动

  查看队列中的任务

  at -l 或 atq

  存放在这里

  ls /var/spool/at/

  删除一次性计划任务

  at -d 序号

  at配置文件，哪些用户可以使用 at

  /etc/at.allow

  /etc/at.deny

    如果存在/etc/at.allow，只有在此文件中的用户才可以使用 at命令

    如果存在/etc/at.deny，在此文件中的用户不可使用at命令

    如果两个文件都不存在，只有超级用户可以使用at命令

    如果两个文件都存在均为空，则所有用户都可以使用 at命令

batch用法与 at一样 [bæ tʃ] n. 一批；一批生产的量；成批作业

  安排在系统负载较轻时运行一次

  一般情况负载较低指 0.8以下(w或top命令能看到)

crontab命令 [krɔn teibl]

  周期性计划任务

  三个常用选项:

  -l 列出当前crontab

  -r 删除当前的crontab 不建议用，直接用-e 编辑就好

  -e 编辑当前的crontab文件

  crontab -e

  分钟 小时 日 月 周 命令/脚本

  把知道的具体时间添上，不知道的用\*号 周日是0

 \* \* \* \* \*

  每天04:00

  0 4 \* \* \* 要执行的命令

  15 4 \* \* \* /usr/bin/php /var/www/html/test.php

  周二和周五 18:00

  0 18 \* \* 2,5 要执行的命令

  1 月到3 月份， 周二和周五18:00

  0 18 \* 1-3 2,5 要执行的命令

  周一到周五 17:30 发广播，然后 17:45 分关机

  写两条计划任务，不是只能写一条

  30 17 \* \* 1-5 /usr/bin/wall < /root/msg

  45 17 \* \* 1-5 /sbin/shutdown -h now

  中午12 点到14点，每两分钟

  \*/2 12-14 \* \* \* 命令

 \* \*/1 \* \* \*

  写命令时，最后要写上出错信息

    例/bin/cp -r /etc /backup/etc.bak 2 >>/backup/etc.bak.err

    2 表示错误信息

  这 6项,都不能为空

  多个值用,

  连续的值用-

  任何时间\*

  命令给出绝对路径

  用户必须具有运行对应命令或程序的权限

  保存在 /var/spool/cron

  检查 crond进程是否开启

  ps -le | grep crond

  /etc/rc.d/init.d/crond start

  有一个配置文件

  /etc/crontab

  定义了执行时的一些环境变量

  最后四行，是一些系统的一些计划任务

    01 \* \* \* \* root run-parts /etc/cron.hourly

    02 4 \* \* \* root run-parts /etc/cron.daily

    22 4 \* \* 0 root run-parts /etc/cron.weekly

    42 4 1 \* \* root run-parts /etc/cron.monthly

  把脚本放在那四个目录中，只要有执行权限，就会被定时执行

  限制哪些用户可以使有 crontab

    /etc/cron.assow

    /etc/cron.deny

service crond restart   重启 crond  重启计划任务

/var/log/message 系统启动后的信息和错误日志，是Red Hat Linux中最常用的日志之一   
/var/log/secure 与安全相关的日志信息   
/var/log/maillog 与邮件相关的日志信息   
/var/log/cron 与定时任务相关的日志信息   
/var/log/spooler 与UUCP和news设备相关的日志信息   
/var/log/boot.log 守护进程启动和停止相关的日志消息

# Linux系统运行

 dmesg命令

 dmesg查看内核引导期间信息

 系统运行级别

0 关机

1 单用户模式

2 不完全多用户模式（不包含NFS服务）

3 完全多用户模式

4 未使用级别

5 图形模式

6 重新启动 系统运行级别

runlevel

查看系统运行级别

init [0-6]

修改系统运行级别 修改系统默认运行级别

vi /etc/inittab、

指定系统默认运行级别。

 ls /etc/rc.d

init.d

rc[0-6].d

rc.local

rc.sysinit

 GRUB

GRUB的配置文件默认为

    /boot/grub/grub.conf

# ls -l /etc/grub.conf （为软链接文件） GRUB配置选项

default 定义缺省启动系统（多系统时才有意义）

timeout 定义默认等待时间（值为0则不等待）

splashimage 定义GURB界面图片

hiddenmenu 隐藏菜单

title 定义菜单项名称

root 定义系统内核所在的分区

kernel 指定内核文件所在位置

initrd 指定镜像文件所在位置 单用户模式应用

   案例：root密码忘记

进入单用户模式重置root密码，方法如下：

开机后进入GRUB界面，按e进入GRUB菜单，再

按e进入菜单项；选中kernel行，再次按e进入编

辑行模式，在行后空格并输入1 ，回车保存后按

b键引导，即可进入单用户模式。

\* 单用户模式不需要密码

passwd root设置新密码，然后exit退出即可 /etc等重要目录目录，装好系统后，做一个备份到/backup/

VMware 启动时按Esc

# SSH

  主要用以下三个功能组件

1.ssh远程登录 Windows下的 SecureCRT、Putty

2.sftp文件共享 类(FTP) SSH Secure File Transfer Client

3.scp 文件共享 类(cp拷贝)

准备环境

  两台电脑

  1.WebServer: 192.168.88.88 这是一台在机房的服务器

  2.MyPC: 192.168.88.99 这是我们自己的工作电脑

Linux下远程登录

  ssh 用户名@地址

  不允许空密码，很多 Linux禁止root远程SSH登录

  选项

    -2 表示使用SSH版本号

    -p 指定端口

  ssh -2 用户名@地址 (建议使用)

  ssh -2 -p 端口号 用户名@地址

  我们在使用 Putty 时，也可以在 Connection中选择使用什么版本

  配置文件:

  vi /etc/ssh/sshd\_config

  Port 22 这是ssh的端口号，默认是注释掉的，去掉#号，就可以改了

  PermitRootLogin no 这样就能禁止root远程登录

  重启服务

  service sshd restart

sftp文件共享

  Linux下命令行：

  sftp 用户名@主机地址

    连上去后可执行 ls get put help bye等

  很少用命令行，我们经常在 Windows下用客户端软件:

    WinSCP(同时支持SCP协议)

    SSH Secure File Transfer Client (安装SSH.Secure.Shell.Client\_3.2.9.exe)

scp 文件共享 (推荐使用)

  本机拷贝到远程主机用法：

  scp 本地文件 用户名@ip地址:远程主机上的目录

  scp -r 本地目录 用户名@远程主机 ip:远程主机上的目录

  scp --help 查看帮助

  例 将 inittab文件传到另一台主机的/root目录中:

  scp /etc/inittab root@192.168.88.88:/root

  从远程主机拷贝到本地:

  scp 用户名@远程主机地址:远程文件 本地目录

  scp -r 用户名@远程主机地址:远程目录 本地目录

  常用选项

  -p 保持原有文件属性 (建议加上)

  -r 复制目录

  -P 指定端口

对称密钥加密 yao

  加密解密使用同一密码

  速度快

  密钥本身需要交换(这就不安全了)

非对称密钥加密

  也称公开密钥加密，会同时生成两个密钥

  一个公开出去(公钥)，一个私人持有(私钥)

  用其中一个密钥加密的数据，只有用另一个密钥才能解密

  公钥加密->私钥解密 (加密文件时使用) 参见“openssl命令行进行RSA加密解密.txt”

  私钥加密->公钥解密 (验证身份，数字签名) 让服务器信任后，使用 SSH时，就不用验证密码了

  非对称密钥加密特点:

  安全性好

  速度慢

scp是需要输入密码的，如何建立信任关系?

  测试环境：

    远程服务器ip为192.168.88.88，用户为webadmin

    本地电脑用户为root

  目的：本机 root登录后，可以直接以 webadmin身份操作远程服务器，不需要验证密码

  原理：让对方(远程服务器)信任我的公钥

  本机建立密钥对:(例如当前登录用户为 root)

  ssh-keygen -t rsa 全部回车即可 RSA 公钥加密算法是1977年由美国麻省理工学院开发的

  会提示生成的公钥 id\_rsa.pub 已生成 ,即是该用户默认的 RSA 身份认证公钥(SSH-2)

  想办法将这个公钥文件拷贝到远程主机上

  scp /root/.ssh/id\_rsa.pub webadmin@192.168.88.88:/home/webadmin

  此时还是需要验证密码的，webadmin是远程主机192.168.88.88 上的普通用户

  到远程主机 WebServer 上，用 webadmin用户登录，找到刚才上传的公钥，设为信任

  ssh -2 webadmin@192.168.88.88

  cd ~ 来到用户宿主目录

  ls -a 如果没有.ssh目录，则创建

  mkdir .ssh

  cat id\_rsa.pub >> .ssh/authorized\_keys #加入到认证信息，用的是追加，可以保存多个信任用户

  chmod 700 .ssh # 新版本的ssh要求这个目录的权限必须是700

  chmod 600 .ssh/authorized\_keys # 新版本的ssh要求这个文件的权限必须是600

  回到 MyPC 本机中，测试

  ssh webadmin@192.168.88.88 不需要密码，直接登录成功

  那么通过 scp拷贝文件，也不需要密码了

  注意:

  在本机中，由于是 root用户下生成的公钥，所以，只要在本机上，用 root用户，以 webadmin身份操作远程主

机，才是信任的

  如果本机不是root用户，或不指定以 webadmin用户操作远程主机，则还是要密码的

  例 ssh root@192.168.88.88 ，尝试以root身份登录远程主机，还是要输入密码

rsync数据镜像备份工具 (remote sync 远程同步) 推荐使用

  WebServer和 MyPC 上都要安装rsync，其中WebServer服务器上是以服务器模式运行rsync，

  MyPC 上则以客户端方式运行rsync。这样在web服务器上运行rsync守护进程，

  在 MyPC上定时运行客户程序来备份 WebServer 服务器上需要备份的内容到MyPC。

  方便的增量备份实现

  可镜像保存整个目录树和文件系统

  保存文件权限、时间、软硬链接等

  文件传输效率高(可压缩传输)

  可以使用 ssh加密通道

  WebServer:

  服务是 xinetd进程托管方式

  默认都已安装，检查是否已安装

    rpm -q rsync

  启动服务 默认不启用

  ls /etc/xinetd.d

  vi /etc/xinetd.d/rsync

  修改为 disable = no 保存退出编辑器

  service xinetd restart 重启xinetd 托管服务

    若提示xinetd: 未被识别的服务，则说明/etc/rc.d/init.d下无 xinetd包

    yum -y install xinetd\* #安装xinetd

    ps -le |grep xinetd #查看是否启动成功

  MyPC:

  在 MyPC中，想要同步服务器上的一个目录到本机测试一下

  rsync -arHz --progress --delete webadmin@192.168.88.88:/website /backup

  注意：webadmin是服务器上的用户，请确定 webadmin有访问服务器/website目录的权限

    本机MyPC 登录用户，要有操作/backup的权限

  如果重复执行，只会操作有改变的文件，如新加的文件或删除的文件，备份目录同步操作

  参数说明

  -a 保持文件属性

  -r 子目录递归处理

  -H 保持文件硬链接 如果要备份的文件中没有硬链接，也可以不加-H

  -z 备份文件传输时压缩处理

  -e ssh 使用ssh加密隧道传输

  --progress 在传输时显示传输过程 可以不加

  --delete 删除目标备份没有的文件

  如果需要将 MyPC中的目录上传到 WebServer，本地目录写在前面即可

实例：

定期将服务器上的/website 目录备份到本机

  每周日 02:00 完全备份，以日期作为目录

  周一到周六 02:00 做一次增量备份

  为什么即要做增量备份，还要做完全备份?

  这里的增量备份，指的是目录同步，跳过相同的文件，如果有文件被删除，则也会删除备份目录中的,所以，需

要定期做完全备份

  crontab -e

  0 2 \* \* 0 /usr/bin/scp -rp webadmin@192.168.88.88:/website /backup/website\_$(date +%Y%m%d)

  0 2 \* \* 1-6 /usr/bin/rsync -arHz --delete -e ssh webadmin@192.168.88.88:/website /backup/web

  取当前日期

    echo $(date +%F) 可查看 date --help

    echo $(date +%Y%m%d)

    给变量赋值:

    # a=5; echo $a;

    # a=(`date +%F`)

    # echo $a

service vsftpd start

匿名用户登录在

/var/ftp

net use \* /del /y //取消连接

net use \\192.168.88.88 "jack" /user:jack

homes

browseable=no 无权限的目录，不显示

valid users=mary jack

      \*\*users\*\*

writable=yes 写权限 yes no

Linux资料 讲师：邹义良 http://www.weibo.com/it266

NFS

Network File System

运行级别2 和3的区别，是否启用了 NFS，运行级别2不启动NFS。

安全性不高

易用性好

让Unix/Linux系统之间的文件共享很简单

最好只在局域网使用

可以像操作本地文件系统一样操作NFS文件系统中的内容

类似于 Windows映射为本地磁盘

可以把远程目录，挂载到本地

查看是否安装了 NFS，需要两个服务:

rpm -q portmap #NFS服务需要用这个软件来分配端口号

rpm -qa | grep nfs #默认都已安装

启动相关服务

service portmap start

service nfs start

检查RPC 程序是否启动

  NFS是通过RPC(Remote Procedure Call,远程过程调用)协议来使用远程计算机上的文件

  所以要检查是否启动了

  rpcinfo -p | more

  能看到有111 端口的 portmapper

配置共享目录

  vi /etc/exports

  格式：

  共享目录 允许访问的主机(权限)

  例：

  /public #没写主机和权限，是允许所有主机只读访问/public目录

  /website 192.168.88.9(ro) #这个 ip只读访问 Linux资料

  /test 192.168.99.\*(ro) #99 网段只读访问

  可写用(rw,sync) 可写访问，并同步保存到硬盘中 (async 资料会先暂存于内存中，而非直接写入硬盘)

  重新启动 nfs服务生效

  service nfs restart

  exportfs -rv 也可以让配置文件生效

查看服务器共享了哪些目录

  showmount -e NFS服务器IP地址

挂载到本地

  mkdir /mnt/dir1

  mount 192.168.88.88:/website /mnt/dir1

取消挂载

  umount /mnt/dir1

  先切换到别的目录(不能坐在凳子上把自己搬起来)

df -h能查看到挂载的目录

NFS访问目录，没有用户控制机制，访问者身份是 nfsnobody 伪用户

  grep nfsnobody /etc/passwd

NFS安全性很差，一般不使用

开机自动挂载NFS共享

  vi /etc/fstab

  192.168.88.88:/website /mnt/dir1 nfs ro 0 0 (最后的两个 0 表示不需要检测，检测顺序为0)

=============================================================================

vsftpd

常用FTP软件

  Wu-FTP:古老、配置复杂

  Proftp:功能强大

  vsftp:安全、高速、稳定 (推荐)

查询

  rpm -q vsftpd

安装

  yum -y install vsftpd

  或

  rpm -ivh vsftpd-2.0.1-5.i386.rpm

启动

  service vsftpd start

  /etc/rc.d/init.d/vsftpd start

  默认启动后，即支持普通用户宿主目录访问及匿名访问

匿名访问

  用户名:

  ftp 或 anonymous

  密码为空或自己的电子邮箱

FlashFXP

文件存在哪里：

  /var/ftp #ftp伪用户宿主目录

windows中连接

  ftp 192.168.88.88

  help查看帮助

  ls

  cd 切换FTP上的目录

  bin 切换到使用二进制传输方式

  lcd 指定下载到本机的哪个目录

  get 下载单个文件

  put 上传

  prompt 关闭交互模式

  mput 上传多个(先关闭交互模式，多个文件用空格分隔)

  mget 下载多个

  bye 退出

  进入到提示行后:

  open ip地址

  user 输入用户名

  建议用客户端软件，自动化时，需要上面的命令行工具

  例：vi auto.ftp

  open 192.168.88.88

  user jack 123456

  bin

  prompt

  lcd /ftp.bak/

  mget \* 下载当前目录所有文件

  bye

  ftp -n <auto.ftp 执行下载 -n是关闭交互模式

配置文件

  vi /etc/vsftpd/vsftpd.conf

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

匿名用户

  anonymous\_enable=YES 是否允许匿名用户访问 关闭：改为 NO

  anon\_upload\_enable=YES 匿名用户允许上传,默认是注释了的

  anon\_mkdir\_write\_enable=YES 匿名用户允许创建目录,默认是注释了的

  anon\_max\_rate=1000000 匿名用户最大传输速率为100KB/s (单位:bytes/sec)

  日志注意三个地方： 尽量要生成日志

  xferlog\_enable=YES 是否使用日志

  xferlog\_std\_format=YES 日志使用标准的格式

  xferlog\_file=/var/log/xferlog 日志文件存放的位置 需要开启

  欢迎信息

  ftpd\_banner=Welcome to blah FTP service.

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

用户访问

  默认普通用户，直接可以登录访问宿主目录

  默认可以上传和下载

  local\_enable=YES

  write\_enable=YES

  local\_umask=022 #创建的目录权限为 755

  local\_max\_rate=200000 用户最大传输速率 为200KB/s

只要系统中的普通用户，默认都能登录 FTP，这样有安全隐患

限制指定用户不能访问，而其他用户可以访问，相关选项

  userlist\_enable=YES

  userlist\_deny=YES

  userlist\_file=/etc/vsftpd/user\_list

例：限制指定用户能访问，其他用户不能访问 建议设置此项

  userlist\_enable=YES

  userlist\_deny=NO

  userlist\_file=/etc/vsftpd/user\_list

  vi /etc/vsftpd.user\_list #一个用户一行

用户默认可以切换到别的目录，只要有rx权限，都能切换过去

  例：ftp> cd /tmp

  250 Directory successfully changed.

chroot (change root) 让 ftp用户把宿主目录当做根目录，这样，就不能切换到别的目录了

  配置文件中:

  chroot\_local\_user=YES

还可以控制哪些用户不受此限制，允许切换到别的有权限的目录

  chroot\_list\_enable=YES

  chroot\_list\_file=/etc/vsftpd/chroot\_list #一个用户一行,列表中的用户，不受chroot限制

其他设置选项

  idle\_session\_timeout=600 空闲10 分钟后被挂断

  max\_clients=50 服务器总并发连接数

  max\_per\_ip=3 每个客户机最大连接数

  listen\_port=10011 指定一个非标准端口 默认连接端口是21，数据传输是20

nmap端口扫描工具

  需要安装 yum -y install nmap

  nmap 192.168.88.88

  telnet 192.168.88.88 22 这样，可以得到 ssh版本，利用漏洞

  nmap默认只扫描知名端口

  nmap -p1-65535 192.168.88.88 扫描全部端口(1到65535)

  《黑客帝国 2》

========================================================

Samba

Windows文件共享

  Windows主机之间使用SMB/CIFS网络协议实现文件和打印资源的共享

  Windows无法与UNIX/Linux 主机之间实现资源共享

Samba服务器

  Samba服务器在UNIX系统中实现了 SMB/CIFS协议

  通过 Samba服务可以实现UNIX/Linux 主机与Windows主机之间的资源互访

特点:

  必须是系统用户，但使用独立密码访问

  默认安装 samba并添加用户(smbpasswd -a 用户名)，即可访问自己的宿主目录(可读写)。

查看是否安装了 samba

  rpm -q samba

  查到 samba-3.0.33-3.28.el5说明已安装

默认安装了客户端:

rpm -qa | grep samba

  samba-client-3.0.33-3.28.el5

  samba-common-3.0.33-3.28.el5

安装

yum -y install samba

启动服务

/etc/rc.d/init.d/smb start

或

service smb start

两个进程:

  smbd ---> tcp 139 用户验证和文件共享

  nmbd --->udp 137 138 浏览共享和计算机名解析

ps -le | grep smbd

ps -le | grep nmbd

用户管理

1.首先必须是系统用户，添加 Linux 系统用户

  useradd 用户名 #添加用户

  passwd 用户名 #设置密码

2.然后设置 samba的密码

  smbpasswd -a 用户名 #加用户

  smbpasswd 用户名 #改 smb密码

  root可以使用管理命令

  pdbedit -a username：新建Samba账户。

  pdbedit -x username：删除Samba账户。

  pdbedit -L：列出Samba用户列表，读取/etc/samba/passdb.tdb 数据库文件。

  pdbedit -Lv：列出Samba用户列表的详细信息。

为了安全，不要把root设为samba用户

重启服务

service smb restart

Windows中如何连接：

  \\服务器地址

  \\192.168.88.88

  可以映射网络驱动器

net use

net use \* /del /y

net use \\192.168.88.88 "password" /user:your\_user

查看有哪些客户端连接

  smbstatus

查看日志

  cat /var/log/samba

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

如果不能访问:

防火墙

  iptables -L 查看

  iptables -F 清空防火墙规则，拆除防火墙

  即时生效，重启后复原

  service iptables stop

  永久性生效，重启后不会复原

  chkconfig iptables off

   chkconfig smb on

SELinux

  允许访问

  setsebool -P samba\_enable\_home\_dirs on

  或完全禁用 SELinux：

  vi /etc/selinux/config 改为SELINUX=disabled

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

如果查找配置文件在哪里

rpm -ql samba-common | grep smb.conf

编辑配置文件

vi /etc/samba/smb.conf

[global] 全局环境选项

  workgroup 工作组或域

  server string 描述信息

  security 安全模式，有如下选项

    share 不验证权限，直接共享

    user 缺省，推荐，由Linux Samba服务器做验证

    server 第三方主机验证

    domain 第三方主机验证(必须是Windows 域控制器)

  hosts allow 哪些主机可以访问(如果没有配置，全部主机都可以)

      如果是ip直接写上，如果是 ip段只写网段部份，多个用空格分隔 如: 192.168.1. 192.168.2.

      哪些主机不允许访问，hosts deny 一般不和hosts allow 同时使用

  log file 日志文件(最好启用日志)

[homes] 只要有这个配置段，用户默认就可以通过自己的用户名，访问自己的宿主目录

  comment 描述

  browseable=no 无权限访问的目录，不显示出来

  writable=yes 可写

[自定义的共享1]

[自定义的共享2]

=================================

配置/software 目录的共享,允许user1 user2两个用户完全访问

创建目录

  mkdir /software

添加用户

  useradd user1

  passwd user1

  useradd user2

  passwd user2

添加samba密码

  pdbedit -a user1

  pdbedit -a user2

如果开启了 SELinux

  chcon -t samba\_share\_t /software

编辑配置文件

  vi /etc/samba/smb.conf

  建议写在配置文件尾

  [software] 共享名，不能有特殊符号，一般是字母或数字组成

  path = /software 共享目录,只能是一个目录

  valid users = user1 user2 多个用空格分开。如果没配置此项，所有用户都可以访问

  writable = yes 写权限 yes no

Samba服务器配置检测命令

  testparm

  如果配置文件写错，例valid users=user1 写成了valid user=user1 将提示

重启服务

  /etc/rc.d/init.d/smb restart

设置目录权限

  直接改为777，但这样不安全

  chmod -R 777 /software

  创建一个用户组，改变目录所属组，对组授权，然后从新连接就可以了

  groupadd softadmin

  usermod -aG softadmin user1

  usermod -aG softadmin user2

  chgrp softadmin /software

  chmod g+w /software

  推荐用 acl

  setfacl -R -m u:user1:rwx /software #让user1 有 rwx权限 (已有目录或文件)

setfacl -R -m u:smb:rwx /webs

setfacl -R -m d:u:smb:rwx /webs

  setfacl -R -m d:u:user1:rwx /software #让以后新创建的文件或目录，user1

也有 rwx权限

  setfacl -R -m u:user2:rwx /software

  setfacl -R -m d:u:user2:rwx /software

windows 下，重新连接

  net use \* /del /y

  net use \\192.168.88.88 "password" /user:your\_user

提供一个共享目录，所有有户都能使用，却不能删除别人的文件

[test]

path = /test

wrieable=yes

然后，设置目录为 chmod 1777 /test

问题：如何在 linux中使用SAMBA 服务器的共享目录？

解答：mount //192.168.88.88/software /mnt/software -o username=your\_user

>输入密码

vsftpd

常用FTP软件

  Wu-FTP:古老、配置复杂

  Proftp:功能强大

  vsftp:安全、高速、稳定 (推荐)

查询

  rpm -q vsftpd

安装

  yum -y install vsftpd

  或

  rpm -ivh vsftpd-2.0.1-5.i386.rpm

启动

  service vsftpd start

  /etc/rc.d/init.d/vsftpd start

  默认启动后，即支持普通用户宿主目录访问及匿名访问

匿名访问

  用户名:

  ftp 或 anonymous

  密码为空或自己的电子邮箱

FlashFXP

文件存在哪里：

  /var/ftp #ftp伪用户宿主目录

windows中连接

  ftp 192.168.88.88

  help查看帮助

  ls

  cd 切换FTP上的目录

  bin 切换到使用二进制传输方式

  lcd 指定下载到本机的哪个目录

  get 下载单个文件

  put 上传

  prompt 关闭交互模式

  mput 上传多个(先关闭交互模式，多个文件用空格分隔)

  mget 下载多个

  bye 退出

  进入到提示行后:

  open ip地址

  user 输入用户名

  建议用客户端软件，自动化时，需要上面的命令行工具

  例：vi auto.ftp

  open 192.168.88.88

  user jack 123456

  bin

  prompt

  lcd /ftp.bak/

  mget \* 下载当前目录所有文件

  bye

  ftp -n <auto.ftp 执行下载 -n是关闭交互模式

配置文件

  vi /etc/vsftpd/vsftpd.conf

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

匿名用户

  anonymous\_enable=YES 是否允许匿名用户访问 关闭：改为 NO

  anon\_upload\_enable=YES 匿名用户允许上传,默认是注释了的

  anon\_mkdir\_write\_enable=YES 匿名用户允许创建目录,默认是注释了的

  anon\_max\_rate=1000000 匿名用户最大传输速率为100KB/s (单位:bytes/sec)

  日志注意三个地方： 尽量要生成日志

  xferlog\_enable=YES 是否使用日志

  xferlog\_std\_format=YES 日志使用标准的格式

  xferlog\_file=/var/log/xferlog 日志文件存放的位置 需要开启

  欢迎信息

  ftpd\_banner=Welcome to blah FTP service.

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

用户访问

  默认普通用户，直接可以登录访问宿主目录

  默认可以上传和下载

  local\_enable=YES

  write\_enable=YES

  local\_umask=022 #创建的目录权限为 755

  local\_max\_rate=200000 用户最大传输速率 为200KB/s

只要系统中的普通用户，默认都能登录 FTP，这样有安全隐患

限制指定用户不能访问，而其他用户可以访问，相关选项

  userlist\_enable=YES

  userlist\_deny=YES

  userlist\_file=/etc/vsftpd/user\_list

例：限制指定用户能访问，其他用户不能访问 建议设置此项

  userlist\_enable=YES

  userlist\_deny=NO

  userlist\_file=/etc/vsftpd/user\_list

  vi /etc/vsftpd.user\_list #一个用户一行

用户默认可以切换到别的目录，只要有rx权限，都能切换过去

  例：ftp> cd /tmp

  250 Directory successfully changed.

chroot (change root) 让 ftp用户把宿主目录当做根目录，这样，就不能切换到别的目录了

  配置文件中:

  chroot\_local\_user=YES

还可以控制哪些用户不受此限制，允许切换到别的有权限的目录

  chroot\_list\_enable=YES

  chroot\_list\_file=/etc/vsftpd/chroot\_list #一个用户一行,列表中的用户，不受chroot限制

其他设置选项

  idle\_session\_timeout=600 空闲10 分钟后被挂断

  max\_clients=50 服务器总并发连接数

  max\_per\_ip=3 每个客户机最大连接数

  listen\_port=10011 指定一个非标准端口 默认连接端口是21，数据传输是20

nmap端口扫描工具

  需要安装 yum -y install nmap

  nmap 192.168.88.88

  telnet 192.168.88.88 22 这样，可以得到 ssh版本，利用漏洞

  nmap默认只扫描知名端口

  nmap -p1-65535 192.168.88.88 扫描全部端口(1到65535)

  《黑客帝国 2》

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

apache 配置

apache的配置文件在哪里?

 cd /usr/local/apache2/etc

 备份这个文件

 cp httpd.conf httpd.conf.20120619

Apache的配置文件修改后，需要重启服务

 /usr/local/apache2/bin/apachectl restart

#号开头的是注释

ServerRoot "/usr/local/apache2/"

 Apache的根目录，配置文件中的相对路径，就是在这个目录下的

Listen 80

 Apache正在监听的端口

 Listen 12.34.56.78:80 #只监听12.34.56.78IP上的80端口

LoadModule rewrite\_module modules/mod\_rewrite.so

 url重写模块

LoadModule php5\_module modules/libphp5.so

 php模块

User daemon

 以daemon用户启动apache进程

Group daemon

ServerAdmin you@example.com

 服务器管理员，会出现在$\_SERVER中

#ServerName www.example.com:80

 服务器的域名和端口，默认不用配置，会自动获取

DocumentRoot "/usr/local/apache2//htdocs"

 文档根目录

 默认网页文件，就存放在这个目录中

<Directory /> #对Linux的根目录进行设置

    Options FollowSymLinks

    AllowOverride None

    Order deny,allow #拒绝优先

    Deny from all #拒绝所有

</Directory>

#对/usr/local/apache2//htdocs目录权限从新配置

<Directory "/usr/local/apache2//htdocs">

    Options Indexes FollowSymLinks

    AllowOverride None

    Order allow,deny #允许优先

    Allow from all #允许所有

</Directory>

<Directory 目录></Directory>用于封装一组指令，只对某一个目录和子目录有效

  Options 选项

  Indexes 列出目录中的文件

  FollowSymLinks 是否支持软链接

#允许所有，拒绝特殊

<Directory "/www">

    Options Indexes

    AllowOverride None

    Order allow,deny

    Allow from all

    Deny from 192.168.88.128

</Directory>

# 拒绝所有，允许特殊

<Directory "/www">

    Options Indexes

    AllowOverride None

    Order deny,allow

    Deny from all

    Allow from 192.168.88.128

</Directory>

<IfModule dir\_module>

     DirectoryIndex index.html index.php #默认文件

</IfModule>

index.html index.asp index.apsx index.php index.jsp delfaut.html defaut..xxxx

#拒绝 .ht开头的文件

<FilesMatch "^\.ht">

    Order allow,deny

    Deny from all

    Satisfy All

</FilesMatch>

#错误日记

ErrorLog "logs/error\_log"

 默认在ServerRoot选项配置的目录下

LogLevel warn

 日志等级:debug, info, notice, warn, error, crit,alert, emerg

DefaultType text/plain

 默认的文件类型

ErrorDocument 404 "/404.html"

 找不到文件，显示什么内容

虚拟主机

 <Directory "/www/qq/">

  Options Indexes FollowSymLinks

  AllowOverride None

  Order allow,deny

  Allow from all

 </Directory>

 <Directory "/www/163/">

  Options Indexes FollowSymLinks

  AllowOverride None

  Order allow,deny

  Allow from all

 </Directory>

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

url重写

www.qq.com/news.php?id=2

www.qq.com/news.php?id=3

www.qq.com/news2.html

www.qq.com/news3.html

设置

AllowOverride All

在目录下创建一个.htaccess文件

 RewriteEngine on

 RewriteRule news(\d+).html new.php?id=$1

====================================================================================================

LINUX关闭防火墙

LINUX关闭防火墙 chkconfig iptables

（1） 重启后永久性生效：

开启：chkconfig iptables on

关闭：chkconfig iptables off

（2） 即时生效，重启后失效：

开启：service iptables start

关闭：service iptables stop

需要说明的是对于Linux下的其它服务都可以用以上命令执行开启和关闭操作。

在开启了防火墙时，做如下设置，开启相关端口，

修改/etc/sysconfig/iptables 文件，添加以下内容：

-A RH-Firewall-1-INPUT -m state ——state NEW -m tcp -p tcp ——dport 80 -j ACCEPT

-A RH-Firewall-1-INPUT -m state ——state NEW -m tcp -p tcp ——dport 22 -j ACCEPT

或者：

/etc/init.d/iptables status 会得到一系列信息，说明防火墙开着。

/etc/rc.d/init.d/iptables stop 关闭防火墙

最后：

在根用户下输入setup，进入一个图形界面，选择Firewall configuration，进入下一界面，选择Security Level为Disabled，保存。重启即可。

======================================================

fedora下 关闭防火墙

/etc/init.d/iptables stop

linux下svn的常用代码

1、将文件checkout到本地目录

svn checkout path（path是服务器上的目录）

   例如：svn checkout svn://192.168.1.1/pro/domain

    简写：svn co

2、往版本库中添加新的文件

  svn add file

   例如：svn add test.php(添加test.php)

   svn add \*.php(添加当前目录下所有的php文件)

3、将改动的文件提交到版本库

  svn commit -m "LogMessage" [-N] [--no-unlock] PATH(如果选择了保持锁，就使用–no-unlock开关)

   例如：svn commit -m "add test file for my test" test.php

    简写：svn ci

4、加锁/解锁

  svn lock -m "LockMessage" [--force] PATH

   例如：svn lock -m "lock test file" test.php

svn unlock PATH

5、更新到某个版本

  svn update -r m path

   例如：

      svn update如果后面没有目录，默认将当前目录以及子目录下的所有文件都更新到最新版本。

     svn update -r 200 test.php(将版本库中的文件test.php还原到版本200)

     svn update test.php(更新，于版本库同步。如果在提交的时候提示过期的话，是因为冲突，需要先update，修改文件，然后清除svn resolved，最后再提交commit)

   简写：svn up

6、查看文件或者目录状态

  1）svn status path（目录下的文件和子目录的状态，正常状态不显示）

   【?：不在svn的控制中；M：内容被修改；C：发生冲突；A：预定加入到版本库；K：被锁定】

  2）svn status -v path(显示文件和子目录状态)

   第一列保持相同，第二列显示工作版本号，第三和第四列显示最后一次修改的版本号和修改人。

   注：svn status、svn diff和 svn revert这三条命令在没有网络的情况下也可以执行的，原因是svn在本地的.svn中保留了本地版本的原始拷贝。

简写：svn st

7、删除文件

  svn delete path -m "delete test fle"

   例如：svn delete svn://192.168.1.1/pro/domain/test.php -m "delete test file"

    或者直接svn delete test.php 然后再svn ci -m ‘delete test file‘，推荐使用这种

简写：svn (del, remove, rm)

8、查看日志

  svn log path

   例如：svn log test.php 显示这个文件的所有修改记录，及其版本号的变化

9、查看文件详细信息

  svn info path

   例如：svn info test.php

10、比较差异

  svn diff path(将修改的文件与基础版本比较)

   例如：svn diff test.php

svn diff -r m:n path(对版本m和版本n比较差异)

   例如：svn diff -r 200:201 test.php

   简写：svn di

11、将两个版本之间的差异合并到当前文件

  svn merge -r m:n path

   例如：svn merge -r 200:205 test.php（将版本200与205之间的差异合并到当前文件，但是一般都会产生冲突，需要处理一下）

12、SVN 帮助

  svn help

svn help ci

——————————————————————————

以上是常用命令，下面写几个不经常用的

——————————————————————————

13、版本库下的文件和目录列表

  svn list path

   显示path目录下的所有属于版本库的文件和目录

简写：svn ls

14、创建纳入版本控制下的新目录

svn mkdir: 创建纳入版本控制下的新目录。

用法: 1、mkdir PATH…

         2、mkdir URL…

创建版本控制的目录。

1、每一个以工作副本 PATH 指定的目录，都会创建在本地端，并且加入新增

     调度，以待下一次的提交。

2、每个以URL指定的目录，都会透过立即提交于仓库中创建。

在这两个情况下，所有的中间目录都必须事先存在。

15、恢复本地修改

svn revert: 恢复原始未改变的工作副本文件 (恢复大部份的本地修改)。revert:

用法: revert PATH…

注意: 本子命令不会存取网络，并且会解除冲突的状况。但是它不会恢复

        被删除的目录

16、代码库URL变更

svn switch (sw): 更新工作副本至不同的URL。

用法: 1、switch URL [PATH]

        2、switch –relocate FROM TO [PATH...]

1、更新你的工作副本，映射到一个新的URL，其行为跟“svn update”很像，也会将

     服务器上文件与本地文件合并。这是将工作副本对应到同一仓库中某个分支或者标记的

     方法。

2、改写工作副本的URL元数据，以反映单纯的URL上的改变。当仓库的根URL变动

    (比如方案名或是主机名称变动)，但是工作副本仍旧对映到同一仓库的同一目录时使用

    这个命令更新工作副本与仓库的对应关系。

17、解决冲突

svn resolved: 移除工作副本的目录或文件的“冲突”状态。

用法: resolved PATH…

注意: 本子命令不会依语法来解决冲突或是移除冲突标记；它只是移除冲突的

        相关文件，然后让 PATH 可以再次提交。

18、输出指定文件或URL的内容。

svn cat 目标[@版本]…如果指定了版本，将从指定的版本开始查找。

svn cat -r PREV filename > filename (PREV 是上一版本,也可以写具体版本号,这样输出结果是可以提交的)

解决冲突

第一种，利用update的选项进行冲突解决，也就是说不管当前拷贝副本是否是最新版本，都使用—accept参数作为冲突处理方式

  --accept ARG : specify automatic conflict resolution action

                            ('postpone', 'base', 'mine-conflict',

                             'theirs-conflict', 'mine-full', 'theirs-full',

                             'edit', 'launch')

(p) postpone - mark the conflict to be resolved later //让文件在更新完成之后保持冲突状态。

(df) diff-full - show all changes made to merged file //使用标准区别格式显示base修订版本和冲突文件本身的区别。

(e) edit - change merged file in an editor //用你喜欢的编辑器打开冲突的文件，编辑器是环境变量EDITOR设置的。

(r) resolved - accept merged version of file //完成文件编辑之后，通知svn你已经解决了文件的冲突，它必须接受当前的内容—从本质上讲就是你已经“解决了”冲突。

(mf) mine-full - accept my version of entire file (ignore their change//丢弃新从服务器接收的变更，并只使用你查看文件的本地修改。

(tf) theirs-full - accept their version of entire file (lose my changes)//丢弃你对查看文件的本地修改，只使用从服务器新接收的变更。

(l) launch - launch external tool to resolve conflict//启动一个外置程序来执行冲突解决，这需要一些预先的准备。

(h) help - show this list //显示所有在冲突解决时可能使用的命令

 /usr/local/nginx/sbin/nginx -s reload

 LNMP状态管理命令：

LNMP状态管理： /root/lnmp {start|stop|reload|restart|kill|status}

Nginx状态管理：/etc/init.d/nginx {start|stop|reload|restart}

MySQL状态管理：/etc/init.d/mysql {start|stop|restart|reload|force-reload|status}

Memcached状态管理：/etc/init.d/memcached {start|stop|restart}

PHP-FPM状态管理：/etc/init.d/php-fpm {start|stop|quit|restart|reload|logrotate}

PureFTPd状态管理： /etc/init.d/pureftpd {start|stop|restart|kill|status}

ProFTPd状态管理： /etc/init.d/proftpd {start|stop|restart|reload}

如重启LNMP，输入命令：/root/lnmp restart 即可，单独重启mysql：/etc/init.d/mysql restart

LNMPA状态管理命令：

LNMPA状态管理： /root/lnmpa {start|stop|reload|restart|kill|status}

Nginx状态管理：/etc/init.d/nginx {start|stop|reload|restart}

MySQL状态管理：/etc/init.d/mysql {start|stop|restart|reload|force-reload|status}

Memcached状态管理：/etc/init.d/memcached {start|stop|restart}

PureFTPd状态管理： /etc/init.d/pureftpd {start|stop|restart|kill|status}

ProFTPd状态管理： /etc/init.d/proftpd {start|stop|restart|reload}

Apache状态管理：/etc/init.d/httpd {start|stop|restart|graceful|graceful-stop|configtest|status}

设置mysql开机自启动

chkconfig mysql on

设置mysql开机不开启

chkconfig mysql off

查看mysql是否开机自启动

chkconfig –list mysql

service mysql stop

chkconfig mysql off

reboot

nginx 的停止和开启

nginx service nginx stop

nginx service nginx start

现在ssh连接的ip地址

方法一：

在/etc/hosts.allow中添加允许ssh登陆的ip或者网段

sshd:192.168.1.1:allow

或者

sshd:192.168.1.0/24:allow

在/etc/hosts.deny添加不允许ssh登陆的IP

 sshd:ALL #ALL表示除了上面允许的，其他的ip 都拒绝登陆ssh

 方法二:

使用iptables:

iptables -A INPUT -p tcp -s 192.168.1.1 --destination-port 22 -j ACCEPT

iptables -A INPUT -p tcp --destination-port 22 -j DROP

方法三：

修改ssh配置文件

vi /etc/ssh/sshd\_config

添加一行：

allowusers xxx@192.168.1.1

注：xxx为你用来登入服务器的用户名。

service sshd restart

nginx允许访问ip限制

location ~ ^/download/ {

 allow 192.168.1.0;

 deny all;

}

// 查看数据库版本 进入mysql -uroot -pyourpwd 然后 mysql> status;

truncate tablename ;主键从1开始

ecos 命令操作

app/base/cmd update minierp

set names gbk

安装openssl

cd /usr/local/src

wget http://www.openssl.org/source/openssl-0.9.8l.tar.gz

tar zxvf openssl-0.9.8l.tar.gz

cd openssl-0.9.8l

./config

make

make install

// 一般apache配置文件

/etc/httpd/conf httpd.conf /usr/local/nginx/sbin/nginx -s reload

 LNMP状态管理命令：

LNMP状态管理： /root/lnmp {start|stop|reload|restart|kill|status}

Nginx状态管理：/etc/init.d/nginx {start|stop|reload|restart}

MySQL状态管理：/etc/init.d/mysql {start|stop|restart|reload|force-reload|status}

Memcached状态管理：/etc/init.d/memcached {start|stop|restart}

PHP-FPM状态管理：/etc/init.d/php-fpm {start|stop|quit|restart|reload|logrotate}

PureFTPd状态管理： /etc/init.d/pureftpd {start|stop|restart|kill|status}

ProFTPd状态管理： /etc/init.d/proftpd {start|stop|restart|reload}

如重启LNMP，输入命令：/root/lnmp restart 即可，单独重启mysql：/etc/init.d/mysql restart

LNMPA状态管理命令：

LNMPA状态管理： /root/lnmpa {start|stop|reload|restart|kill|status}

Nginx状态管理：/etc/init.d/nginx {start|stop|reload|restart}

MySQL状态管理：/etc/init.d/mysql {start|stop|restart|reload|force-reload|status}

Memcached状态管理：/etc/init.d/memcached {start|stop|restart}

PureFTPd状态管理： /etc/init.d/pureftpd {start|stop|restart|kill|status}

ProFTPd状态管理： /etc/init.d/proftpd {start|stop|restart|reload}

Apache状态管理：/etc/init.d/httpd {start|stop|restart|graceful|graceful-stop|configtest|status}

设置mysql开机自启动

chkconfig mysql on

设置mysql开机不开启

chkconfig mysql off

查看mysql是否开机自启动

chkconfig –list mysql

service mysql stop

chkconfig mysql off

reboot

nginx 的停止和开启

nginx service nginx stop

nginx service nginx start

现在ssh连接的ip地址

方法一：

在/etc/hosts.allow中添加允许ssh登陆的ip或者网段

sshd:192.168.1.1:allow

或者

sshd:192.168.1.0/24:allow

在/etc/hosts.deny添加不允许ssh登陆的IP

 sshd:ALL #ALL表示除了上面允许的，其他的ip 都拒绝登陆ssh

 方法二:

使用iptables:

iptables -A INPUT -p tcp -s 192.168.1.1 --destination-port 22 -j ACCEPT

iptables -A INPUT -p tcp --destination-port 22 -j DROP

方法三：

修改ssh配置文件

vi /etc/ssh/sshd\_config

添加一行：

allowusers xxx@192.168.1.1

注：xxx为你用来登入服务器的用户名。

service sshd restart

nginx允许访问ip限制

location ~ ^/download/ {

 allow 192.168.1.0;

 deny all;

}

// 查看数据库版本 进入mysql -uroot -pyourpwd 然后 mysql> status;

truncate tablename ;主键从1开始

ecos 命令操作

app/base/cmd update minierp

set names gbk

安装openssl

cd /usr/local/src

wget http://www.openssl.org/source/openssl-0.9.8l.tar.gz

tar zxvf openssl-0.9.8l.tar.gz

cd openssl-0.9.8l

./config

make

make install

// 一般apache配置文件

/etc/httpd/conf httpd.conf

mysql 永久关闭 修改文件 /etc/rc.local 写代码 /etc/init.d/mysqld stop

然后还要输入命令

service mysqld stop

chkconfig mysqld off

系统重启后 mysql就不会再自动启动了

安装links

 yum -y install links

**一、**安装crontab

**[root@CentOS ~]# yum -y install vixie-cron  
[root@CentOS ~]# yum -y install crontabs**

查找目录或文件 命令

find / -name  phpmyadmin

2、赋予root权限方法一： 修改 /etc/sudoers 文件，找到%wheel一行，把前面的注释（#）去掉## Allows people in group wheel to run all commands%wheel ALL=(ALL) ALL然后修改用户，使其属于root组（wheel），命令如下：#usermod -g root tommy修改完毕，现在可以用tommy帐号登录，然后用命令 sudo su - ，即可获得root权限进行操作。方法二： 修改 /etc/sudoers 文件，找到root一行，在root下面添加一行，如下所示：## Allow root to run any commands anywhereroot ALL=(ALL) ALLtommy ALL=(ALL) ALL修改完毕，现在可以用tommy帐号登录，然后用命令 sudo su - ，即可获得root权限进行操作。

方法三： 修改 /etc/passwd 文件，找到如下行，把用户ID修改为 0 ，如下所示：tommy:x:500:500:tommy:/home/tommy:/bin/bash修改后如下tommy:x:0:500:tommy:/home/tommy:/bin/bash保存，用tommy账户登录后，直接获取的就是root帐号的权限。

建议使用方法二，不要轻易使用方法三。

当找不到命令行输入php -v找不到的时候，可以如下执行

在/usr/bin 目录下建立软链接连接到php的安装目录 首先 cd /usr/bin 然后  例如我的 执行命令 ln -s /phpstudy/server/php/bin/php  php

groups fxh  查看 用户fxh属于哪些组

crontab 详解

  & 后台执行命令

       当在前台运行某个作业时，终端被该作业占据；而在后台运行作业时，它不会占据终端。可以使用&命令把作业放到后台执行。

       如：

       30 2 \* \* \* /data/app/scripts/hotbackup/hot\_database\_backup.sh &

       在后台运行作业时要当心：需要用户交互的命令不要放在后台执行，因为这样你的机器就会在那里傻等。

       不过，作业在后台运行一样会将结果输出到屏幕上，干扰你的工作。如果放在后台运行的作业会产生大量的输出，最好使用下面的方法把它的输出重定向到某个文件中：

       如：

              command >out.file 2>&1 &

       在这个例子中，2>&1表示所有的标准输出和错误输出都将被重定向到一个叫做out.file 的文件中。

2.4  2>&1 含义

先看一个例子：

0 2 \* \* \* /u01/test.sh >/dev/null 2>&1 &

这句话的意思就是在后台执行这条命令，并将错误输出2重定向到标准输出1，然后将标准输出1全部放到/dev/null 文件，也就是清空。

在这里有有几个数字的意思：

       0表示键盘输入

       1表示标准输出

       2表示错误输出.

我们也可以这样写：

0 2 \* \* \* /u01/test.sh  >/u01/out.file &  --这里没写，默认是1

0 2 \* \* \* /u01/test.sh  1>/u01/out.file &

0 2 \* \* \* /u01/test.sh  2>/u01/out.file &

0 2 \* \* \* /u01/test.sh  2>/u01/out.file  2>&1 &

将tesh.sh 命令输出重定向到out.file, 即输出内容不打印到屏幕上，而是输出到out.file文件中。

2>&1 是将错误输出重定向到标准输出。 然后将标准输入重定向到文件out.file。

&1 表示的是文件描述1，表示标准输出，如果这里少了&就成了数字1，就表示重定向到文件1。

& ：后台执行

测试：

ls 2>1 ： 不会报没有2文件的错误，但会输出一个空的文件1；

ls xxx 2>1： 没有xxx这个文件的错误输出到了1中；

ls xxx 2>&1： 不会生成1这个文件了，不过错误跑到标准输出了；

ls xxx >out.txt 2>&1 == ls xxx 1>out.txt 2>&1；  因为重定向符号>默认是1，这句就把错误输出和标准输出都传到out.txt 文件中。

2.5  2>&1写在后面的原因

       格式：command > file 2>&1   == command  1> file 2>&1

       首先是command > file将标准输出重定向到file中， 2>&1 是标准错误拷贝了标准输出，也就是同样被重定向到file中，最终结果就是标准输出和错误都被重定向到file中。

如果改成： command 2>&1 >file

       2>&1 标准错误拷贝了标准输出的行为，但此时标准输出还是在终端。>file 后输出才被重定向到file，但标准错误仍然保持在终端。

useradd –g ftp –s /sbin/nologin ftpuser  
新建用户加入ftp组，  
/sbin/nologin意思是用户不允许登录（通过ftp可以连接）  
用户名为ftpuser

nginx 禁止直接访问目录配置

在server里面

location /{

      deny all;

      return 403;

 }