ULE - 112

2009 7 29

一. Linux 简介

1. Linux 版本命名格式为 xx.yy.zz，其中 xx 表示主版本号，yy 表示次版本号，zz 表示修订版本号
2. 主版本号升级比较慢，目前只启用了 1 和 2 两个版本
3. 次版本号使用奇数代表开发版本，偶数代表稳定版本。

二. Linux 目录结构

树型，即在整个系统中只有在一个根目录（文件系统），其他都挂载在根目录下相应的子目录节点。

/ ：根目录
/boot ：用于存放启动所必需的文件
/var ：用于存放系统中经常需要变化的一些文件，如系统日志
/home ：用于存放所有普通用户的宿主目录
/root ：系统管理员 root 的宿主目录
/bin ：用于存放系统基本用户命令
/sbin ：用于存放系统基本的管理命令
/usr ：用于存放系统大量的应用程序
/etc ：用于存放系统和各种程序的配置文件
/dev ：设备文件
/lib ：库文件，驱动
/tmp ：临时文件
/media ：光盘的默认载入点，
/mnt ：软盘的默认载入点，
/misc
/proc ：虚拟文件系统，它放置的数据都在内存中，所以本身不占任何硬盘空间
/sys ：文件系统目录，
/opt ：给主机额外安装软件所放的目录
/srv ：一些服务启动之后，这些服务所要访问的数据目录

三. Linux 文件系统

1. EXT2 和 EXT3，EXT3 属于日志文件系统
2. swap 类型的文件系统在交换分区中使用虚拟内存，大小设置为内存的 1.5 — 2 倍
3. 大多数 Linux 系统支持 FAT32 文件系统的读写和 NTFS 的只读

No.

Date

四. 分区设备

IDE硬盘

hda　a表示第1个　　a-d

hda1　1表示第1个分区　　1-63

SCSI硬盘

sda　a表示第1个　　a-p

sda1　1表示第1个分区　　1-15　最多4个主分区 或 3个主分区1个扩展分区

五. 安装

/	EXT3	30G
/boot	EXT3	200M
/home	EXT3	1G
	swap	2G

六. 热键

ctrl + alt + F1-6　　#图形界面切换到字符界面

alt + F7　　#字符界面切换到图形界面

ctrl + shift + +　　#放大窗口字符大小

ctrl + -　　#缩小窗口字符大小

ctrl + shift + t　　#新建窗口

alt + 数字　　#切换窗口

ctrl + l　　#清屏

shift + pageup　　#上翻

ctrl + C　　#中断进程

=

七. 简单常用命令

1. date　　#日期

2 cal　　#日历

cal [08] 2009

3. history　　#历史命令

4. mii-tool　　#查看网卡是否连接正常

5. service network restart　　#重启网络服务　/etc/init.d/network restart

6. ftp 192.168.1.254 ↵

:~> ls

cd pub ↵

ls

minor blues ↵　或 get+文件名

7. shutdown -r now　　#重启

8. shutdown -h now　　#关机

script　　#记录所有操作，保存为文件 typescript

No.

Date 2009 · 7 · 30

一. 帮助

1. --help /-h

ls --help

2. man + 命令　　158　　#手册

man 1 cat　　# cat命令基本参数

man 5 passwd　　# passwd文件格式

man -a　　#所有

man -f　　# man文件列表

man -k passwd　　#查询含有"passwd"的帮助文件

3. info +命令　　# GNU的超文本帮助系统

info date　　进入 menu下的定义

ctrl +h　　# 热键列表

Backspace　　# 返回上一级

n　　# 跳到下一小节

p　　# 跳到上一小节

u　　# 本章总目录

s<字符串>　　# 查找

4. /usr/share/doc/　　# 功能描述文档

例: cd /usr/share/doc/dnsmasq-2.45

ls

less setup.html

5. howto　　# 只对某个事情的操作文档

6. google

二. 简单文件管理命令

(一) 查

1. ls　　#查看目录中的对象

ls -a　　#查看目录中所有文件包括隐藏(.开头)

ls -l　　#显示当前目录里对象的详细信息, ll

ls -t　　#以时间顺序排列查询结果

ll -h　　#显示大小

ls -d　　#查看当前目录信息.

ls -i　　#显示 inode 号

ls -r　　#逆向排序

ls -R　　#递归显示目录结构

2. cat + 文件名　　#查看文件内容, 只显示最后页, 适合小文件

3. more　　#分屏显示, 翻到最后直接退出无法前翻

4. less　　# ... , 可前翻

space: 下页　b: 上页　↓: 下行　q: 退出

G. end 直接到最后

g. home　　前

查找关键字 :

? 光标当前往回之前 / 之后　　vim

n: 下一个关键字　shift+n: 上一个

5. head　　#默认显示前10行

head -n 20 或 head -20　　#显示前20行

head -n -20　　#不显示最后20行

6. tail　　#默认显示最后10行

tail -f　　#实时监控　　echo "12345" >> /etc/passwd

7. rev　　#左右颠倒

8. tac　　#前后颠倒

9. pwd　　#显示当前所在目录路径

10. du　　#显示当前目录中对象的大小

du -s　　#显示当前目录大小

11. tree　　#树形显示目录

12. cd ..　　#回到父级目录

cd -　　#回到上一次目录

cd ~ 或无任参数　　#回到当前用户主目录

(二) 建

1. mkdir　　#创建一个空目录

mkdir a b c　　#创建同级别空目录 a b c

mkdir -p a/b/c　　#创建层次空目录

No.

Date

touch file　　#更新文件访问时间

2. touch file1 file2　　#新建指定文件名的空文件

touch -d 20071115 install.log　　#修改显示时间　Access、Modify

1130

touch -t　　#修改详细时间　man touch

(三) 复制、移动、删除、

1. cp file原 file新　　#将当前目录中的file原复制为file新

cp file a　　#将当前目录中文件file复制到当前目录子目录a中

cp -r　　#复制目录及里面内容

cp -f　　#强制覆盖

2. mv file原 file新　　#源文件与目标文件在同一目录时移动等同于重命名

mv file a　　#将当前目录中文件file移动到当前子目录a中

-i　　#提示

-v　　#显示过程

3. rmdir　　#删除一个空目录

rmdir a b c　　#删除同级别空目录 a b c

rmdir -p a/b/c　　#删除一个空目录，若删除后上一级目录也为空则一并删除

4. rm file1 file2　　#同时删除文件file1, file2

rm -r　　#删除目录

rm -f　　#强制删除

(四) 编辑

1. gedit　　#图形界面

2. vim　　#打开编辑器

insert　　#输入

esc , : wq　　#保存退出

q!　　#不保存

(五) file + 文件名　　#查看文件属性

find + 路径 + 条件　　#查找

find . -name "filex"　　#按照文件名进行查找

三.用户管理

(一)添加

1. useradd + 用户名　　#创建用户
2. passwd + 用户名　　#给用户创建密码
3. useradd -u uid 用户名　　#创建用户时指定用户ID
   useradd -g gid 用户名　　#创建用户时指定主组
   useradd -G gid,gid 用户名　　#创建用户时同时将用户加入一些候选组
   useradd -c 姓名 用户名　　#创建用户时添加说明信息
   useradd -s /sbin/nologin　　#创建此用户不能登录

(二)修改

1. usermod -u uid 用户名　　#修改用户ID
   usermod -g gid 用户名　　#修改用户主组
   usermod -G gid,gid 用户名　　#将用户同时加入一些候选组
   usermod -l 新用户名 原用户名　　#修改用户名
   usermod -L 用户名　　#禁用用户帐号
   usermod -U 用户名　　#启用
   usermod -e YYYY-MM-DD 用户名　　#修改帐号有效期限

(三)删除

1. userdel + 用户名　　#删除帐号
   userdel -r　　#删除用户帐号同时删除用户的宿主目录

(四)切换用户

1. su - 用户名　　#切换用户
2. exit　　#返回
3. id　　#查看用户uid,gid及所属组.

(五)涉及文件

1. 超级用户　root　uid=0
   系统帐号　　1-499
   普通帐户　　500-60000
2. /etc/passwd　　#保存着系统中所有帐号,所有用户都可读

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

x: 密码　0: uid　0: gid　root: 说明信息　/root: 主目录　/bin/bash: 登录运行的第一个程序

man 5 passwd

No.

Date

3. /etc/shadow　　#保存用户口令，只有root和管理员可读取

root:　　:14454:0:99999:7:::

　　↑
密码(!!空)

(十六) 查看用登录信息

1. whoami
2. who
3. w
4. users　　#查看当前登录用户

2009 7 31

一、组管理

(一)添加

1. groupadd + 组名　　#创建组
   groupadd -g gid 组名　　#创建组时指定组ID
2. gpasswd + 组名　　#给组设置密码
   gpasswd -a 用户名 组名　　#将用户加入组
   gpasswd -d 用户名 组名　　#将用户从组中删除
   -A 用户,用户

(二)修改

1. groupmod -g gid 组名　　#修改组ID
   groupmod -n 新组名 原组名　　#修改组名

(三)删除

1. groupdel 组名　　#删除组

(四)切换组身份

1. newgrp 组名　　#切换组身份(暂时)
2. exit　　#返回

例1: useradd -G root aa　　#新建用户aa同时把aa加入组root中
id aa　　#查询aa所属组 aa, root
用aa登录
newgrp root　　#暂时切换aa组身份为root组
id　　# gid=0(root) 已切换
exit　　#返回aa身份

例2: useradd bb　　#新建用户bb
gpasswd root ↵ 123　　#给root组设置密码123
用bb登录
newgrp root　　#切换组身份,提示输入密码123
id　　# gid=0(root) 已切换
exit　　#返回

No.

Date

(五)涉及文件

1. /etc/group　　#保存所有组帐号信息

root : x : 0 : root

密码　组ID　成员

2. /etc/gshadow　　#保存组口令

root : : : root

密码　管理组　成员

二、创建用户流程

1. /etc/passwd
2. /etc/shadow
3. /etc/group
4. /etc/gshadow
5. mkdir /home/username
6. cp /etc/skel/* /home/username
7. chown username:username /home/username

三、创建用户影响的文件

1. /etc/passwd
2. /etc/shadow
3. /etc/group
4. /etc/gshadow
5. /etc/login.defs　　策略控制　pass_max_days, gid_min
6. /etc/skel

四 1. last　　#查看用户登录日志

last +用户名/控制台

last log　　#帐户最后登录情况，默认所有用户

2. finger +用户名　　#查看用户信息

3. write root pts/1 ↵　　#给root发送信息

给谁发　控制台

4. tty　　#查询控制台
5. wall "hello."　　#广播
6. mesg n　　#关机拒绝接收
7. logout　　#退出登录的shell
8. exit　　#退出shell

五、权限管理

(一)基本权限

1. -|rw-|r--|r--
   文件类型　属主权限(u)　属组权限(g)　其他用户权限(o)

2. -普通文件　d目录文件　l链接文件 ln　b块设备文件 U盘
   c字符文件 键盘　p管道文件　s套接字文件 /tmp

3. r:读 4　w:写 2　x:执行 1

4. +:增加相应权限　-:减少相应权限　=:赋予相应权限

5. chmod u+x file　　#设置文件权限
   g-w
   o=x
   a　xw
   chmod 644 file

6. chown user file　　#设置文件属主
   chown :g file　　#设置文件属组　:或.
   chown user:g file　　#同时设置文件属主和属组
   chown -R　　#同时改变子目录权限

7. chgrp group file　　#设置文件属组

No.

Date

(二)权限列表

			/bin/			touch mkdir	
	cat	vim	./		ls	rm	cd
	r	w	x		r	w	x
f_---				d_---			
f_--x				d_--x			✓
f_-w-				d_-w-			
f_-wx				d_-wx		✓	✓
f_r--	✓			d_r--	?		
f_r-x	✓		✓	d_r-x	✓		✓
f_rw-	✓	✓		d_rw-	?		
f_rwx	✓	✓	✓	d_rwx	✓	✓	✓

1. 用root帐号新建用户test密码123
2. 进入/home/test 新建文件f_，目录d_
3. 修改f_，d_属主为test　　chown -R test /home/test
4. 修改属主权限同文件名，目录名，属组和其他用户权限都为0
   chmod 000 f_---
   chmod 100 f_--x
5. 在d_r--，d_rw-下新建文件123
   touch /home/test/d_r--/123
6. 用test帐号登录测试 · 注：test可以删除f_---，因为test对自己主目录有rwx权限

(三)高级权限

1. 权限掩码　/etc/bashrc　root中 ↓　普通用户 ↓

普通文件起始权限 666 → 644　664　r

目录文件 … 777 → 755　x

2. umask　#查询权限掩码

root umask 0022 —二进制→ 000 010 010 —取反→ 111 101 101

uid 0-99　普通文件起始权限 666 —二进制→ 110 110 110

相与

644 ←十进制— 110 100 100

普通用户 umask 0002 —二进制→ 000 000 010 —取反→ 111 111 101

110 110 110

相与

664 ←十进制— 110 110 100

3. suid —— 设置使文件在执行阶段具有文件所有者的权限

(命令的执行取决于命令所有者的权限而非执行人的)

-rws r-x r-x　例. ll /bin/ping　-rwsr-xr-x 1 root

例：$ cat /etc/shadow　无法查看

chmod u+s /bin/cat

ls /bin/cat -l　-rwsr-xr-x 1 root root …

$ cat /etc/shadow　可以查看

4. sgid —— 针对普通可执行文件，必须应用程序，且有x权限，以拥有者决定(同 suid)

针对目录，必须对目录有 wx 权限，内部新文件将有和目录相同组权限

5. sticky —— 文件其他用户即使有写权限也无法删除

例：# mkdir /ttt

touch /ttt/file

chmod 777 /ttt

$ rm /ttt/file

chmod o+t /ttt

$ rm /ttt/file 无法删除

其他用户可删除，权限777的目录/ttt下的文件，但不是自己创建的.

No.

Date

6.

suid	sgid	sticky	
0	0	1	1 普通目录文件
1	0	0	4 应用程序 例 /usr/bin/passwd 4755

例：# mkdir /ttt

chmod 1777 /ttt

六. 文件结构介绍 inode + block = EXT3

1. ln -s 原文件名 快捷方式 #创建符号连接

例：ln -s /sbin/cat C

./C /etc/passwd = cat /etc/passwd

ln 原文件名 新文件名 #创建硬连接（防误删除）

2、新建普通文件，硬连接数为1

新建目录文件，硬连接数为2，目录文件的硬连接数=2+子目录数

3. ls -lia #显示inode号

硬连接：指向同inode号的一个名字，不能跨分区

2009 8 3

一、高级文件管理

1. grep 过滤

```
grep 'root' /etc/passwd        #过滤含有'root'字符串的行
grep '^root' /etc/passwd       #过滤以'root'字符串开头的行
grep 'bash$' /etc/passwd       #过滤以'bash'字符串结尾的行，锚定行尾
                               △查询可登录的用户（整行显示）
grep -c 'bash$' /etc/passwd    △#统计数字只
grep -r 'passwd' ./            #过滤当前目录下含有'passwd'的文件
  -n                           # 显示行号
  -m +数字                     #允许输出的行数
```

2. diff 比较

例1：

```
#vim a.txt
q
a
z
w
s
x
```

```
#vim b.txt
q
a
z
-
w
s
x
```

例2 new

```
a:        b:
q         q
a         a
z         z
w         +w
s         s
x         x
```

```
#diff a.txt b.txt
```

结果：3a4

例1 > -

7d7

<

例2 4c4

< w

> +w

#比较a.txt和b.txt的不同

a：多的，后一个文件比前一个多

> ：后面比前面多的内容

d：少的，后一个文件比前一个少

< ：前面比后面多的内容

c：发生不同的行

No.

Date

3. uniq 合并 (相邻的)

例: #vim uniq.txt
a
a
a
c
c
y

uniq uniq.txt #合并相同的行
a
c
y

uniq -c uniq.txt #合并相同的行并显示合并数统计重复的行
3 a
2 c
1 y

例2: vim uniq.txt
a
a
a
c
a
c
y

uniq uniq.txt
a
c
a
c
y

uniq -c uniq.txt
3 a
1 c
1 a
1 c
1 y

uniq -u #显示没有重复的行
-d #显示重复的行
-c #统计重复的行

4. cut 截取段

例: cut -d: -f1 /etc/passwd #截取 /etc/passwd 第一段用户名
△ 查询所有用户名
△ -d #指定分隔符
△ -f #段号
-c #显示某个字符到某个字符范围

cut -c1-5 /etc/passwd

5. sort 排序（根据ASII码）

例：# vim sort.txt

a
b
d
c
1
11
2
4
5

sort sort.txt

1
11
2
4
5
a
b
c
d

#排序

例2 #sort -n sort.txt

a
b
c
d
1
2
4
5
11

#排序（0一字母一数字）

sort -r

-u

-f

△ -t

△ -k

△ -n

#降序

#去掉重复的行

#忽略大小写

#指定分隔符

#段号

#数字排序

例3. sort -n -t: -k3 /etc/passwd

#从第3段以序排序
按uid升序排序

No.

Date

6. wc 统计

```
wc -l /etc/passwd        # 统计行        △共多少个用户，只数字
   -c                    # 统计字节
   -m                    # 统计字符
   -w                    # 统计单词
```

二. VIM 编辑器

(一) 命令模式

1. 进入输入模式

```
i        #在当前光标前进入插入状态，
I        #  当前光标所在行首
a        #  当前光标后
A        #  当前光标所在行末
o        #  当前行下面
O        #  当前行上面
s        #  删除当前光标所在字符并进入插入模式
S        #              行
```

2. 复制

```
yy       #复制当前整行
yw       #复制单词（包括空格）
ye       #复制单词（不包括空格）
```

3. 剪切

```
dd       #剪切当前整行
dw       # 剪切单词
```

4. 粘贴

```
p        #粘贴到当前光标后，如果复制整行则粘贴到下一行，
P        #          前，              上一行，
```

8.删除
dd
r
x

#替换当前光标处字符
#删除当前光标处字符并退格

5.撤消
u
U
ctrl+r

#撤消上一次操作
#撤消所有操作
#对撤消进行恢复

6.方向
k上
h左 l右
j下

G
gg

#跳转到文件末尾行
#首

7.可视 v. 光标选择区域，复制 y，剪切 d

(二)末行模式：
1.退出
:q!
:wq
:w /tmp/aa
2:e!
3:r !ls -l
:r filename
4:set nu
:set nonu
5.:nohl
6.:help
7.:!命令
8.查找替换
:/word
:?word
n
N

#不保存退出
#保存退出
#另存为
#重编辑
#读取参数中指定的文件并将内容粘贴到当前行下面
#显示行号
#不显示行号
#关闭高亮度显示
#
#从上向下查找
#自下向上
#下一个
#上一个

:1,$ s/old/new/g　　#从首行到末行全部替换　.表示当前行
　　1和$可以换成任意行号，

:%s/old/new/g　　#整篇文档替换

9. :syntax on/off　　#颜色开关

(三) 1. vim -o a.txt b.txt　　#同时编辑a.txt和b.txt，上下分屏显示

vim -O a.txt b.txt　　#　　左右分屏

ctrl+w w　　#切换编辑窗口

qa　　#全部退出

2. /etc/vimrc　　#内容键定义件

三、查找

1. which 例查应用程序命令

which cat　　#显示 /bin/cat

which ls　　#显示 alias ls='ls --color=tty'
　　/bin/ls

#PATH

which cp /rm　　#显示 alias cp='cp -i'
　　/bin/cp
　　i:提示

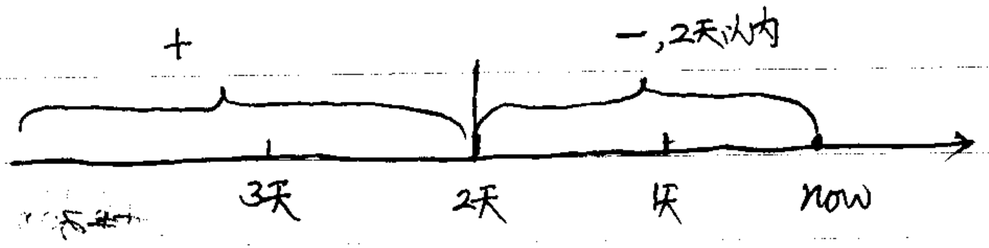
2. whereis cat　　#显示 /bin/cat
　　/usr/share/man/man1/...
　　手册位置

3. locate passwd　　#在整个硬盘中查找
updatedb　　#更新数据库
anacrontab
　　每天04:02更新

例. locate nmap

4. find

```
find /etc -name passwd          # 指定目录下查找文件名是"passwd"的文件
                                  如不指定目录则在当前目录下查找
          -iname                # 不区分大小写
/home     -user blues           # 按用户
          -uid 500              # 按 uid
          -gid 500              # 按组ID
          -group blues          # 按组
          -inum 4577571         # 按 inode号
          -links 4              # 按硬连接数
          -type f               # 按文件类型 7种
          -size 10M             # 按大小   -:小于   +:大于
          -perm 644             # 按权限644
                -644            # 有效位必须满足,其他位可有可无.
                +644            # 有效位有一个满足即可      rw-r--r--
access    -amin 30              #按访问时间(分钟)
          -atime 2              #          (天)
modfy     -mmin                 #按修改内容时间(分钟)
          -mtime +2             #          (天) 2天以外
change    -cmin                 #按修改属性时间(分钟)
          -ctime -2             #          (天) 2天以内修改
```



\; 执行的次命令分隔符

```
          -exec 命令 {} \;      #对找到的结果显示进行处理
find /ttt -uid 500 -exec rm {} \;
                    ok          #提示是否操作
find /etc -type f -mtime -1 -exec ls -lh {} \;
find /usr -perm +4000           #查找设置了suid的可执行文件
find /var -size 10M -exec ls -lh {} \;
```

No.

Date

四. 打包压缩

(一)压缩 gz . bz2

1. gzip + 文件名　　#压缩
   gunzip + 文件名.gz　　#解压缩
   -f　　#最快
   -b　　#最好

2. bzip2 + 文件名　　#压缩
   bunzip2 + 文件名.bz2　　#解压缩
   -1　　#最快
   -9　　#最好

3. zip + 文件名　　#压缩
   unzip + 文件名.zip　　#解压缩

(二)打包 tar cf 文件名.tar 目录/文件 —— 需要备份的文件或目录

c　　#创建归档文件
r　　#追加
t　　#查看
x　　#释放
v　　#显示过程
z　　#gz压缩
j　　#bz2压缩
-C　　#指定目录（释放时）
cvf　　#建立
xvf　　#解开
rvf　　#追加
tvf　　#显示过程
zcvf　　#建立gz

zxvf　　#解压gz
jcvf　　#建立bz2
jxvf　　#解压bz2

例：tar cf etc.tar /etc　　#创建归档文件
tar rf etc.tar /boot　　#追加
tar tf etc.tar　　#查看
tar xf etc.tar -C /tmp　　#释放（释放后原包仍存在

(三) 打包压缩

tar zcf 文件名.tar.gz 需打包的文件名　　#创建压缩归档文件 gz
zxf　　-C /tmp　　#解压缩
ztf

tar jcf 文件名.tar.bz2 需压缩的文件名　　#创建压缩归档文件 bz2
jxf

No.

Date 2009 8 4

一. 重定向与管道

(一) 输入输出

ls -l /dev/std*

1、输出 >

/dev/stdin → /proc/self/fd/0		标准输入	键盘
/dev/stdout	1	标准输出	显示器
/dev/stderr	2	错误输出	

>	#输出重定向，覆盖原来内容，相当于 1>
>>	#输出重定向追加，不覆盖原来内容
2>	#错误输出
2>>	#错误输出追加
&>	#正确、错误一起输出
>> 2>>	#正确、错误一起输出追加

例：ls -l > txt　　#将ls -l查看的内容输出到txt中，cat txt查看

例2：ls -l txt 000　　#不存在000，正确、错误信息一起输出

结果：ls: 000: No such file or directory

-rw-r--r-- 1 root ----

ls -l txt 000 > txt　　#正确信息输出到txt中.

结果：ls: ---

ls -l txt 000 2> txt　　#错误信息输出到txt中

结果：-rw-r--r-- ---

ls -l txt 000 &> txt　　#正确、错误信息都输出到txt中

ls -l txt 000 >> txt 2>> txt　　# ------ 追加到txt中

例3：tty　　#查看当前窗口(控制台)号

ls -l > /dev/pts/2　　#输出到另一窗口

ls -l txt 000 >> /dev/pts/2 2>> /dev/pts/2　　#正确、错误信息都追加到另一窗口.

例4. ls -l txt 000 &> oo.txt　　#一个管道

ls -l txt 000 > oo.txt 2>&1　　#把错误信息转成正确的强行输出

两个管道

2、输入 <

cat ↵　　#从键盘获取并显示

cat < /dev/zero -A　　#从文件中输入，-A 有显字符变为可显

/dev/zero 零

ls -l > /dev/null　　# /dev/null 黑洞，垃圾桶

cat <<EOF ↵　　# 打开文档缓冲区，输入并显示

> 输入字符 ↵

> EOF

例1: lftp 192.168.1.254 <<EOF　　#下载

> get pub/blues/blues.ncd

> EOF

例2: vim download.sh　　#下载：第1步，编辑、保存

#!/bin/bash

lftp 192.168.1.254 <<EOF↵

get pub/blues/blues.ncd↵

EOF

:wq

chmod +x download.sh　　=　　第2步，修改权限，增加可执行权限

-x，u,g,o 三位都加

第3步，执行下载到当前目录

./download.sh

No.

Date

(二) 管道

1、 命令1 | 命令2 | 命令3　　前一个命令的输出作为后一个命令的输入

stdin → stdin

stdout → stdout

stderr → stderr

例1. grep 'bash$' /etc/passwd | wc -l #统计可登录的用户数

例2. grep 'bash$' /etc/passwd | cut -d: -f1　#截取可登录的用户名

2. tee复制数据流到文件

例：cat /etc/passwd | grep 'bash$' | tee 11.txt | grep '^r'

cat 11.txt　　#检测

3. tr. 删除字符

例：ls -l | tr -d '\n' | cat　　#删除回车

二. 进程管理

(一) 查

1. ps

```
ps a
ps au
ps aux
ps auxw
ps auxf
ps e
ps -eo "%p %c"
```

```
当前        pts/1   shell
#查看进程控制台 tty1  字符界面
#查看控制台进程
#显示用户 -u
#显示不在控制台的进程,完整 -x
#宽屏输出 -w
#全格式输出 -f   family tree
#显示所有进程
#指定输出格式 PID和命令 -o
```

man ps

%C	pcpu	%CPU
%G	group	GROUP
%P	ppid	PPID
%U	user	USER
%a	args	COMMAND 参数及命令
%c	comm	COMMAND
%g	rgroup	RGROUP
%n	nice	NI 优先级
%p	pid	PID PID号
%r	pgid	PGID
%t	etime	ELAPSED
%u	ruser	RUSER 真实用户ID
%x	time	TIME
%y	tty	TTY 控制台号
%z	vsz	VSZ 虚拟内存

+ — 当前进程组

STAT：系统状态　S-sleeping休眠　s-session当前会话　R-run运行

%MEM：内存使用率　RSS物理内存　START：开始时间

No.

Date

2. pstree | less　　#分屏显示进程树
pstree -p | less　　#显示PID号

3. pgrep　　#过滤进程号
例：pgrep -u root sshd　　#过滤 root用户的sshd进程号
sshd—进程名

4. top ↵　　#时时监控进程，每2s刷新1次
>< 翻页
enter 刷新　　zombie 僵尸进程、孙

(二)控
top：i、r ↵ 输入PID号 ↵ -20-19　　#在监控状态按下r，修改进程优先级
-20 ~ 19 共40个数字，-20优先级最大，
nice—优先级，Linux分时操作系统，
优先级越高获得使用CPU的时间段越长

2. k ↵ 输入PID号 ↵ 1,9,15　　# 15—正常结束进程 TERM
9—kill，init kill不掉 KILL
1—重置、挂起、中断 HUP

3. nice -n 19 sleep 500　　#以优先级19运行命令休眠500s
nice -n -20 cat　　设置优先级

4. renice -20 -p 3209　　#调整PID号是3209的进程的优先级是-20

5. kill 3204　　#结束PID号是3204的进程，不加参数默认15
kill -9 3204　　#等同于 kill -KILL 3204

6. killall -TERM Httpd —进程名　　#结束进程组

7. kill -l , killall -l

例. pstree -p | less

httpd 3474 { httpd(PID号)

kill -HUP 3474 , pstree -p | less # HUP, 1, 重置.

httpd 3474 { httpd(新PID号)

(三).

/proc/* # !重要,不能随意改动.

less /proc/cpuinfo # 查看CPU信息.

less /proc/meminfo # 查看内存信息.

例: echo c > /proc/sysrq-trigger 死机

ls /proc # 映射当前运行的进程的PID号

No.

Date

三、后台任务

1. command&　　#命令后加&直接在后台运行该命令，当前控制台输出

sleep 5008　　#休眠5008，[1]后台任务号

2. jobs　　#查看后台运行的任务

+：当前后台进程，-：次当前后台进程

3. kill %1　　#终止后台任务，%1：任务号为1.

4. ctrl + z　　#后台任务终止

bg %1　　#激活后台终止的进程

5 fg %1　　#调到前台

6. disown %1　　#删除后台任务（从后台列表中移除并没有终止）

四、系统状态查看

1. df -h　　#查看文件系统

2. du -h /root　　#显示/root内容的大小

du -sh /root　　#显示/root大小

/home：du --max-depth=1 -h | sort -n　　#统计当前目录下深度一层的目录和文件的大小并升序排列.

用户

3. stat + 文件名　　#文件信息查看（大小，硬连接数，访问时间等）

-f　　#文件所在文件系统的信息.

4. free　　#查看内存

-m　　#以兆为单位

5. uname　　# 显示系统信息，　Linux

　　-a　　# 显示系统全部信息，-r 内核版本号

6. hostname　　# 主机名

7. lspci
   lsusb　　# 显示总线设备信息，

8. dmesg　　# 内核启动信息

9. x86info　　# 详细 CPU 信息，

10. dmidecode　　# 主板信息，

No.

Date

五. BASH相关命令

用户
shell
kernel
硬件

ls /bin/*sh

结果：/bin/bash csh ksh sh tcsh zsh

(一)

1. alias c='cat' #设置别名

unalias c #取消别名

2. history ↙ #历史命令

!1015 #调用命令历史

!! #调用最后一个命令

!$ #调用最后一个命令的最后一个参数

!string #调用命令历史中倒数第一个匹配字符串的命令

例：ls -l date #date

Ls

(二) 变量介绍及操作

set #当前系统中全部变量

env #环境变量 env | less

例：echo $SHELL # /bin/bash，查看环境变量

结果：/bin/bash

echo SHELL

结果：SHELL

echo \$SHELL

结果：$SHELL

常用环境变量（子进程继承父进程）

$PWD　　#当前工作路径
$HOME　　#用户主目录
$PATH　　#命令的路径
$OLDPWD　　#上一次操作过的目录
$USER　　#当前用户名
$LOGNAME　　#帐户登录名
$MAIL　　#邮件预设路径
$HOSTNAME　　#主机名
$HISTSIZE　　#历史记录最大数
$INPUTRC　　#需要载入的资料
$PS1

(三) 登录配置文件

1. /etc/profile　　#全局(用户)配置文件，任何用户登录都加载此文件
　　设置环境变量
　↓
　/etc/profile.d/*　　#存放脚本目录，不同shell环境执行不同脚本
　　SHELL → .sh , CSHELL → .csh
　/etc/bashrc　　#全局BASH的配置文件

2. ~/.bash_profile　　#用户配置文件(只加载一次)
　~/.bashrc　　#用户BASH的配置文件(多次加载)　shell环境

3. ~/.bash_logout　　#退出时执行的文件

4. ~/.bash_history　　#存储的历史命令

登录：/etc/profile → /etc/profile.d/* → ~/.bash_profile → ~/.bashrc
　→ /ect/bashrc

非登录：~/.bashrc → /etc/bashrc → /etc/profile.d/*　?

No.

Date

例1. alias c='cat'　　放到 ~/.bash_profile　#只在登录时加载一次，重新登录有效

~/.bashrc　#每开一个窗口都重新加载

ls -l .bash*

例2. vim ~/.bashrc 在最后添加 alias d='date'

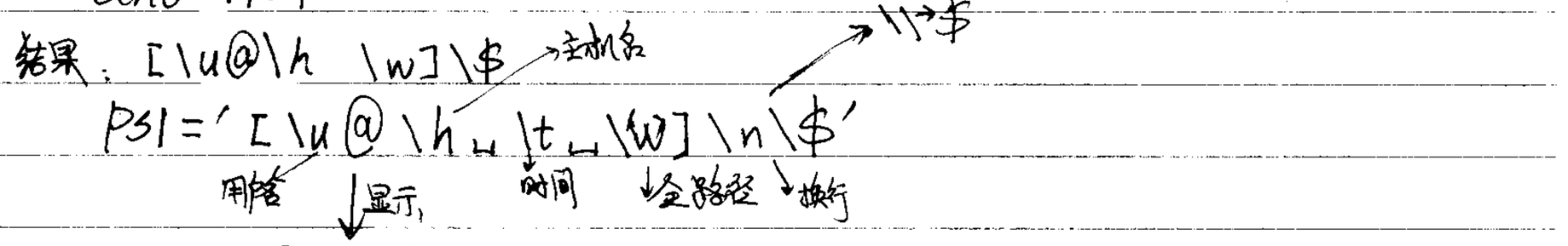
bash ↙　　#新开一个环境

d ↙　　#等同于 date

例3. man bash → /PS1 变量

echo $PS1

结果：[\u@\h \w]\$



将此 PS1=… 命令修改到 ~/.bashrc　#用户登录或新开 bash 环境都做此改变

vim ~/.bashrc　添加到最后一行 # Source global definitions 前

(四) BASH符号

~	#主目录
$	#变量替换
%	#后台任务操作符
&	#转入后台执行，宏替换符
*	#任意1个字符，当前目录下的所有文件 ls /bin/*sh
[ ]	#指定一个字符范围 ls /dev/sda[234]；ls /dev/sda1[2-4]
{x,y,z}	#枚举 ls /dev/sda{10,15,25}

|

\

' '

" "

;

?

#管道

#转义(脱)echo \$SHELL 显示$SHELL字符串

echo '$SHELL $PWD' 强脱义符

弱脱义符，对$,!不起作用

#命令分隔符

touch txt11;cat txt11;chmod +x txt11

#匹配一个字符的位置

ls /dev/tty?

结果：/dev/tty0 /dev/tty1-9

No.

Date 2009 8 5

软件包安装

源代码安装

复制安装（拷贝、二进制、绿色软件）

包管理器 RPM Linux, Unix

APT debian, unbuntu

YAST SUSE使用

一. RPM包管理器

安 rpm -ivh 包名 #安装RPM文件，i:install，v过程，h:hash进度条

rpm -ivh --test 包名 #检测依存关系

rpm -ivh --nodeps 包名 #忽略依存关系安装

rpm -ivh --force 包名 #强制安装

查 rpm -q 包名 #查询包是否安装

rpm -qa #查询系统中所有已安装的包

rpm -qa | wc -l # 统计所有包个数

rpm -qa | grep mysql #查询包名含有字符串 mysql 的包

rpm -ql 包名 | less #查看包文件列表

rpm -qf /bin/ls #反查ls是哪个包安装的

rpm -qi 包名 #查看包的详细信息

rpm -qR 包名 #查询已安装包的依存关系

例 rpm -qf /usr/bin/passwd

passwd-0.73-1

rpm -qi passwd-0.73-1

rpm -qR passwd-0.73-1

No.

Date

```
rpm  -qpl 包名      #查询未安装包的文件列表
rpm  -qpi 包名      #查询未安装包的详细信息
rpm  -qpR 包名      #查询未安装包的依存关系
```

删：
```
rpm  -e 包名            #删除已安装的RPM包
rpm  -e --nodeps 包名   #删除，对下载依存关系
```

二. YUM管理器

服务端：

```
createrepo  /var/ftp/pub/RHEL5U3        #创建依存关系图文件，产生 repodata 目录
cd repodata
ls
```

- primary.xml.gz : rpm信息标记文件，输出信息相当于 rpm -qpl，还包括包之间的依存关系
- filelists.xml.gz : rpm包内文件列表标记文件，输出信息相当于 rpm -ql，rpm -qR
- other.xml.gz : 标记其他信息如版本号
- Comps.xml : 包组信息

客户端

```
vim /etc/yum.repos.d/server1.repo
[server1]  仓库名
name=Red Hat Enterprise Linux 5 Update 1   仓库描述
baseurl=ftp://192.168.1.254/pub/RHEL5U1    网络路径
enabled=1   1仓库开启，0库包关闭
gpgcheck=0  不进行 hash 检查
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat
```

```
yum install 包名        #装包（装包时先下载服务器里的3个文件）
```

Server

例: mount /dev/cdrom /mnt

mkdir /var/ftp/pub/RHEL5U3

cp -rf /mnt/* /var/ftp/pub/RHEL5U3

find / -name createrepo* #查找包名中含有createrepo的包

rpm -q createrepo-0.4.11-3.el5.noarch.rpm #查询此包是否安装

rpm -ivh createrepo-0.4.11-3.el5.noarch.rpm #安装此包

createrepo /var/ftp/pub/RHEL5U3

Client

vim /etc/yum.repos.d/server1.repo

[server1]

name=my yum server

baseurl=file:///var/ftp/pub/RHEL5U3

enabled=1

gpgcheck=0

gpgkey=file:///etc/pki/rpmgpg/RPM-GPG-KEY-redhat-release

yum install cman #安装

yum search cman #查找是否安装

yum list #显示装载了的包列表,包括库中

yum update 包名 #升级

yum info 包名 #显示包详细说明

---

cp -r /mnt/* /var/ftp/pub/RHEL5U3

chmod +x bluesyum.sh

./bluesyum.sh

---

yum whatprovides 文件名 #仅查

yum localinstall 包名 #装本地rpm包

yum localupdate 包名 #本地升级

三. 源码安装 Source

```
tar zxf 源码.tar.gz -C /usr/local/src

./configure --prefix=路径        #检测操作系统所具有的工具生成Makefile说明书
make                             #编译代码为二进制文件
make install                     #拷贝文件到具体目录中
make uninstall                   #卸载
make clean                       #清除编译的文件
```

```
cd /root/software
```

例1：源码安装 notecase

```
tar zxf notecase-1.9.8_src.tar.gz -C /usr/local/src
cd /usr/local/src/notecase-1.9.8
./configure
make
make install
```

例2：安装 office

```
tar zxf EIOffice-Personal-Lin.tar.gz -C /usr/local/src
cd /usr/local/src/EIOffice-Personal-Lin
ls
chmod +x setup.sh
./setup.sh
```

例3：安装英汉辞典

```
tar jxf dic.tar.bz2 -C /usr/local/src
mv /usr/local/src/dic /usr/share/stardict
```

例4：安装虚拟机

```
./VMware-Workstation-6.5.1-126130.i386.bundle
```

ULE - 115

No.

网络配置

Date 2009 · 8 · 6

一. 网络接口

1. 设备名称.

eth0 , eth1　　#第一块网卡，第二块网卡

lo　　#本地环回测试，供本地进程间通信

PPP 0　　# adsl 拨号状态下看到的设备

FDDI　　#光纤

2. 显示网络接口命令

ifconfig　　#显示所有网卡配置，不包括 down

ifconfig -a　　#包括 down

lspci | more ether　　#查看以太网相关信息.

lspci | grep -i ether --color　　#查看硬件网卡，:后是网卡厂商

ifdown eth0　　#禁用网卡

ifconfig eth0 down　　#禁用网卡

ifup eth0　　#启用网卡

ifconfig eth0 10.0.0.1 up　　#启用网卡并设IP地址

二. 为网卡配置 ip　　/etc/rc.d/rc.local 万能配置文件，启动执行，将命令添加至此文件

1. ip = 网络号 + 主机号

子网掩码由连续的1加连续的0组成，子网掩码的1有多长，ip网络位有多长，分隔网络

2. 配置ip地址命令 临时

ifconfig eth0 192.168.1.1 netmask 255.255.255.0　#设置ip

ifconfig eth0 192.168.1.1/24

ifconfig eth0:0 192.168.1.1/24　　#为一块网卡绑定虚拟ip

eth0:1

3. 命令配置网关 临时

route add default dev eth0 gw 默认网关地址　#添加默认路由

route add -net 目标网段 netmask 255.255.255.0 dev eth0 gw 下一跳

192.168.4.0　#添加静态路由

route add -host 10.0.0.3 netmask 255.255.255.255 dev eth0 gw 下一跳

#发往主机的路由

route del　#删除路由

4. 修改网卡配置文件设置 ip·网关，重启生效，永久

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

5. system-config-network-tui 相当于 neat-tui　#重启生效，永久

/etc/init.d/network restart　相当于 service network restart

6. 开启包转发功能

echo 1 > /proc/sys/net/ipv4/ip-forward　#开启包转发功能，临时

vim /etc/sysctl.conf　#永久，修改第一行 net.ipv4.ip_forward = 1

三.

1. 查看路由表

route -n　#-n表示不解析主机名，

netstat -r　路由表按精确度排优先级

子网掩码产生路由的优先级

2. 查看端口

netstat -antp　#查看本机开放哪些tcp端口

netstat -anup　#　　udp

netstat -i　#查看接口 interface、RX接收，TX发送

multicast 多播，txqueuelen 发送队列长度

collisions 冲突

No.

Date

nmap -sT 对方IP地址　　#扫描对方开放哪些tcp端口，不加-s默认tcp

nmap -sU 对方IP地址　　#　　udp

nmap -sP 192.168.1.0/24 -n　　#扫描网段中的所有IP能否ping通

四. 主机名

1. hostname　　#查看主机名

hostname 新主机名　　#修改主机名，临时

2. vim /etc/sysconfig/network　　#修改主机名称配置文件，重启生效，永久

vim /etc/hosts　　#修改主机名同时修改主机名称解析文件，使主机名与127.0.0.1对应

127.0.0.1　localhost.localdomain　localhost

192.168.1.1　www.163.com　　级别高于DNS

五. DNS 设置

1. vim /etc/resolv.conf　　#修改域名服务器配置文件，

search　localadmain　　ping www 时会search后的域名

nameserver　DNS服务器IP

nameserver　192.168.1.100

nameserver　192.168.1.200

2. host 主机名　　#查看主机对应的IP地址

dig 主机名　　#　　〃

nslookup　　#域名解析

六. DHCP

→ DHCP Discover

← DHCP offer

→ DHCP request

← DHCP ACK

七. telnet

1. 客户机服务器模式 C/S
2. 服务器开启23端口, tcp
3. 输入服务端用户名密码
4. 普通用户登录 su - root
5. telnet ip

telnet 127.0.0.1 25 #指定登录端口

helo aa.com

mail from: root@aa.com

rcpt to: root

data ↵ 输正文 . 退出

mail ↵ #查看邮件, 输编号

八. ssh.

1. 加密
2. 客户机服务器模式 C/S
3. 服务器开启22端口
4. 默认开启
5. ssh 192.168.1.254 #以当前身份登录, 打开shell

ssh 192.168.1.254 "ls -l"

ssh 普通用户@远程ip #普通用户登录 su - root

6. /etc/init.d/ssh stop #关闭ssh
7. scp [普通用户@]远程ip:/tmp/abc /本地路径 拷贝远程文件到本地

scp /本地路径 [普通用户@]远程ip:/tmp/abc 拷贝本地到远程

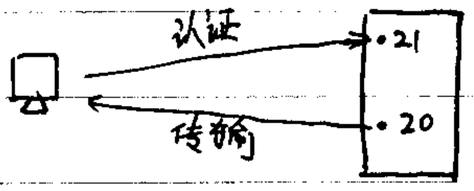
No.

Date

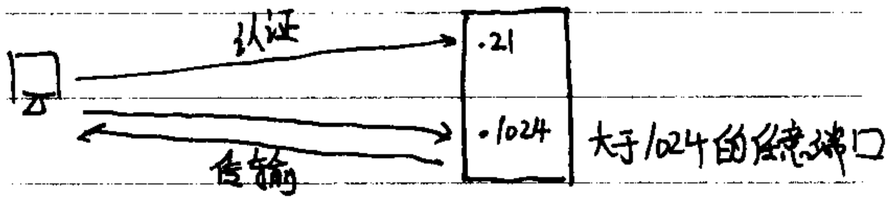
九. ftp

1. 客户机服务器模式 C/S

2. 主动



被动



3. ftp 服务器ip 匿名用户名ftp,密码任意

4. lftp 直接匿名

5. get , mget *.sh #下载

put , mput #上传

图形界面下：应用程序 → 系统工具 → 文件浏览器 → 文件 → 连接到服务器
→ 输入服务器ip → 连接

十. 网页浏览

firefox

elinks http://

#字符界面浏览器

十一. wget /var/www/html/passwd

wget -r -p -k -np http://192.168.1.254/passwd

#-r：递归下载

-np：不下载别的站点链接

-p：获得所显示网页所需元素，pic. flash

-k：将下载的网页链接改为本地链接

-l：下载几层目录

-c：断点续传

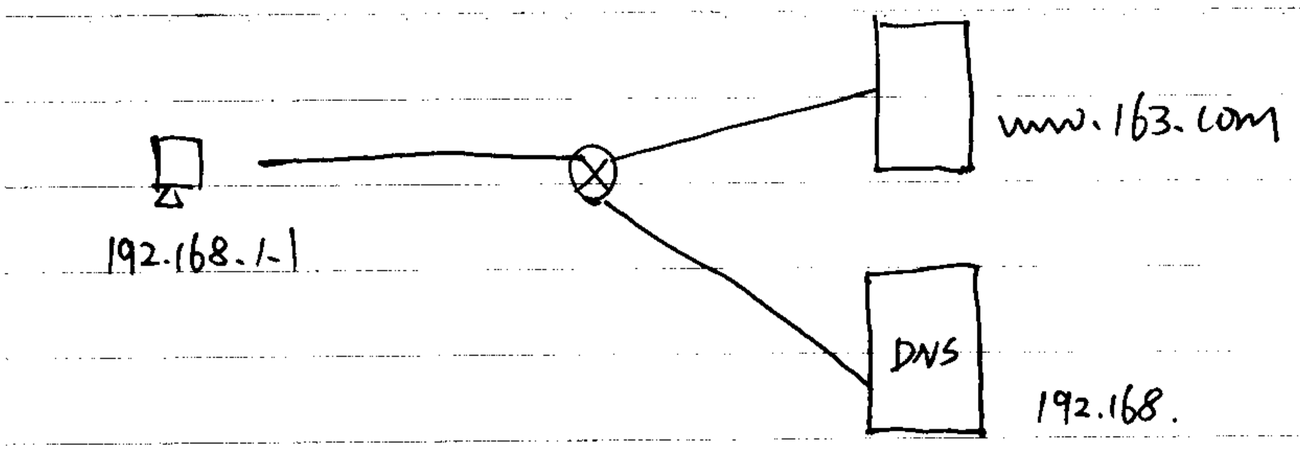
-t 100：重试100次

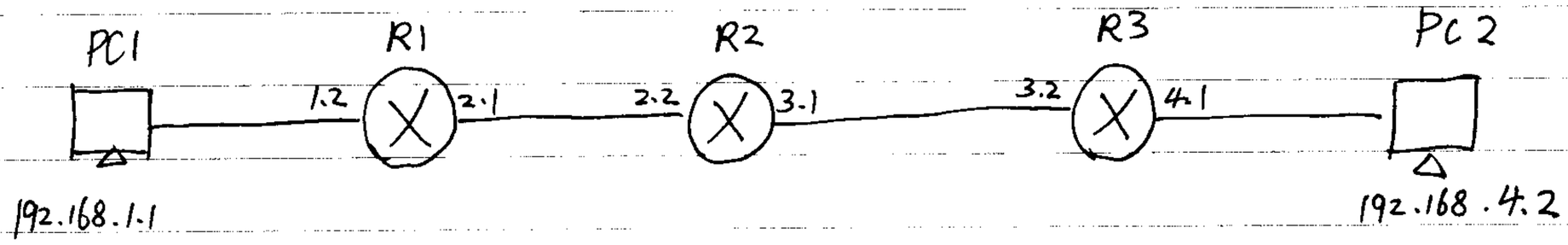
-t 0：无穷次重试

十二. 路由跟踪

traceroute 192.168.1.217 -n #查看经过多少路由器，等

mtr 192.168.1.217





	打开包转发	打开包转发	打开包转发	
设IP 1.1	设IP 1.2，2.1	net 4.0，下一跳 3.2	设IP 3.2，4.1	设IP 4.2
网关 1.2	默认路由 2.2	net 1.0，下一跳 2.1	默认路由 3.1	网关 4.1

十三. 排错方法

1. mii-tool　　#查看网卡连接状态.
2. ping lo
3. ping 自己ip
4. ping 网关
5. ping dns
6. ping www
7. 浏览器

```
ping -c 4 192.168.1.1      #指定ping的次数
ping -f 192.168.1.1        #洪水ping
ping -s 3000               #指定包大小3000字节
```

No.

Date

私有IP网段

10.0.0.0/8

172.16.0.0/12

192.168.0.0/16

169.254.0.0/16

127.0.0.0/8

多播地址

255.0.0.1 —— 234.255.255.255

共 $2^{32}=4$ G 个IP　　G, T, P

CIDR无类间路由

nat网络地址转换

ISO组织的OSI模型7层

TCP/IP模型5层，便于管理

应用层			telnet	DNS, DHCP
传输层	段		port	端到端通信，tcp（可靠，面向连接），udp（不可靠，面向无连接，无序
网络层	包	路由	ip	点到点通信，网络间通信 ↓arp
链路层	帧	交换	mac	网络拓扑结构 ethernet（总线型，高速），PPP（点到点，树型，安全）
物理层	比特			

kickstart

例：服务端

```
yum install system-config-kickstart
system-config-kickstart                          #保存生成 ks.cfg文件
cp /root/anaconda-ks.cfg 自己选项到ks.cfg中

mount /dev/cdrom /mnt
mkdir -p /var/ftp/pub/RHEL5U3
cp -rf /mnt/* /var/ftp/pub/RHEL5U3               #安装树
mv /root/ks.cfg /var/ftp/pub/RHEL5U3

vim /etc/exports                                 #nfs共享目录
    /var/ftp/pub        192.168.1.0/24 (ro,sync)
    service nfs restart
    showmount -e (192.168.1.254)                 #查看本机是否共享
    chkconfig nfs on                             #下次引导启动nfs
```

客户端

```
mount 192.168.1.254:/var/ftp/pub /mnt
```

插入光盘输入命令：linux ks=nfs:192.168.1.254:/var/ftp/pub/ks.cfg
http

通过DHCP获得IP.

2009 8 7

一、分区

```
fdisk -l                                  #查看分区信息，相当于parted /dev/sda print
fdisk -l /dev/sda                         #查看此设备的分区信息      df -h
                                                                    df -i inode剩多少

fdisk /dev/sda                            #打开fdisk交互界面
      n                                   #新建分区. p主. e扩展. l逻辑分区
      +1000M                              #设定大小
      d                                   #删除分区，输入分区号
      p                                   #打印分区表
      l                                   #列出分区格式对应编号
      t                                   #修改分区系统ID 83
      w                                   #保存退出
      q                                   #退出，
partprobe /dev/sda                        #立即生效、同步
```

二、文件系统（不同文件系统的效率取决于应用）

1、inode

block

superblock

mbr

```
#存放属性信息，128字节，stat查看，从2开始
#存放文件内容 1，2，4k，对于目录，存放着目录下
 各的文件的inode
#存放inode，block信息，
 每份区最前面，指示信息
#在整个磁盘最前面，存放分区表
```

2. EXT3是日志文件系统，保持数据的一致性，在文件系统检测时加快速度

```
dumpe2fs /dev/sda1 | less                 #文件系统信息
    filesystem state      clean           #读写改
    inode count                           #inode总数
    block count
    mount count           18              #已挂载次数
/etc/fstab { maximum mount count  -1      #最大挂载次数，超过将进行磁盘检查
检测的条件 { check interval                -1永不自检
                                          tune2fs -c 修改
```

inode block
1:4

mke2fs -j -b 2048 -i 8192 /dev/sda7

-i：平均1个inode占空间大小

3. 创建文件系统，格式化

#创建ext3文件系统

mkfs.ext3 设备名/dev/sda1

-b 1024 ：指定block大小，默认4096

ext2

-L ：创建文件系统同时指定label

msdos = fat

=mke2fs -j /dev/sda7

vfat = fat32

cramfs

tune2fs -j 设备名 #将ext2转换为ext3格式文件系统，日志，

-l #查看文件系统信息 数据不丢失

-c #强制自检的挂载次数

4. parted /dev/sda print #查看分区信息

parted /dev/sda mkpart logical ext3 12.7GB 12.8GB

#创建分区，立即生效，12.7和12.8GB为开始结束位置

parted 设备 rm 分区号 #删除分区

三. 挂载 -t ext3可省略

1. 挂：mount -o 参数,参数2 设备名称 挂载点 #要使用一个分区必须先挂载，挂载点必须是目录

rw, ro

async, sync #异步，同步

auto, noauto #自动

dev, nodev

suid, nosuid

exec, noexec #执行execute

nouser, user #mount权有root可以执行，使用user参数一般用户也能挂载

remount #重新挂载，系统debug或更新参数时

defaults默认值为：rw, async, auto, dev, suid, exec, nouser

CPU
cache
RAM
硬盘
异步

2. 查：mount #查看挂载了哪些文件系统

cat /etc/mtab #..

对文件系统做修改要先卸载

3. 卸：umount 设备名或挂载点

fuser 挂载点 # 卸载分区系统

ps -p 进程号 # busy无法卸载时，查看正在运行的进程号

C表示当前目录

※ 4. /etc/fstab # 开机自动挂载文件，mount -a

LABEL=/home	/home	ext3	defaults	1	2.
别名	挂载点，	文件系统格式	默认参数	dump备份软件	开机检查顺序
			若此处只写user		0不检查
tmpfs 进程间通信	虚拟分区		default其他参数都保留		挂载次数
devpts 虚拟终端					启动时间
sysfs 存放hal硬件抽象层					检测一致性
proc 存放内核映象					

e2label /dev/sda5 /home # 设置别名，卷标名，最长16字节

设备名 别名

e2label 设备名 # 查看此设备的别名

mount LABEL=/home 相当于 mount /dev/sda5，不能直接 mount /home

四、ACL访问控制列表

1. dumpe2fs -h /dev/sda1 | grep acl --color # 查看文件系统中acl信息

mount -o remount,acl 设备名 挂载点 # 重新挂载使支持ACL

2. setfacl -m u:用户名:rx acl_test # 给用户设置acl权限

-m g:组名： # 组

getfacl acl_test # 查看文件ACL权限

setfacl -x u:用户名 acl_test # 删除文件ACL权限

3. setfacl -m m:r acl_test # 设定mask，相与，effective？

setfacl -b acltest # 删除所有ACL属性

4. setfacl -m d:u:用户:rx /tmp/a #预设权限

原来存在于此目录中的文件 file1 权限不变

新建文件 file2 继承此目录权限

copy来的文件 file3 继承此目录权限

mv来的文件 file4 保持原来权限

五. attr 设备级权限，root也受限制

chattr +i acl_test #代表文件不可被改变，相当于 mount -o ro

-i #取消

chattr +a acl_test #代表文件只能追加内容 >>，日志文件推荐

-a #取消

lsattr #查看

chattr +

-

=

作业：

1. 使优盘开机自动挂载

vim /etc/fstab

/dev/sdb1 /mnt vfat defaults 0 0

/dev/cdrom /mnt iso9660 defaults 0 0

1分钟

2. 在a用户主目录中写脚本 "shutdown -h 10"，使b用户进入a主目录执行此脚本关机

useradd a passwd a 123

useradd b passwd b 123

vim /home/a/shutdown.sh #!/bin/bash ↵ /sbin/shutdown -h 10

setfacl -m u:b:rx /home/a

setfacl -m u:b:rx /home/a/shutdown.sh

chmod u+s /sbin/shutdown (which shutdown)

或 setfacl -m d:u:b:rx /home/a 再在a目录下写脚本

或PATH="$PATH:/sbin"，这样脚本中不必写绝对路径（" "表示追加，' '表示赋值）

No.

Date 2009 · 8 · 11

一. 磁盘配额

1. 支持磁盘配额功能

vim /etc/fstab　　#修改 /etc/fstab文件

/boot　defaults, usrquota, grpquota　　#修改权限添加用户,组磁盘配额功能

mount -o remount /boot　　#重新挂载

2. 建立数据库文件

quotacheck -cug /boot　　#产生aquota.user, aquota.group 两文件

3. 打开磁盘配额功能

quotaon /boot　　#开启

quotaoff /boot　　#关闭

4. 编辑　-g

例1. edquota -u zorro ↵

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/sda1	已占用	警告	禁止	已占用	警告	禁止
	以K为单位	延迟天		以个为单位		

↓

edquota -t 修改警告延迟时间以分为单位

8000　10000　10

dd </dev/zero > file

du -h file

touch {1,2,3,4}{a,b,c,d}

例2. 给10个帐户建立相同磁盘配额.

edquota -p zorro user{1,2,3,4,5,6,7,8,9,10}

复制谁的　复制给谁

for i in 'seq 1 10'; do useradd user $i; done

#创建10个用户

例3. cut -d: -f1 /etc/passwd | tail -11 | xargs echo

setquota

quota -u 用户名　　#显示用户配额情况

-g

No.

Date

二、系统启动引导

1. bios —— 主板 —— 基本输入输出系统 —— 引导操作系统

↓ 设置

2. mbr (recorder) —— 硬盘 —— 分区表 —— 主引导记录 —— 0号柱面 8M

512字节

446字节	MBL(code) 主引导代码
64字节	DPT 硬盘分区表 每个字节标记1个分区，最多4个
2字节	55AA 标记结束

MBL(code) → bootloader 启动加载器

chainloader 多系统 +1 代表切换另一个 bootloader

Linux：grub

↓ 配置文件

/boot/grub/grub.conf ※

/boot/grub/menu.lst：grub.conf 的符号连接

stage1：mbr 引导，嵌入 mbr 后面

stage2：读取

vim /boot/grub/grub.conf

↓ 加载

```
default=0                                  #开机默认启动第1个系统
timeout=5                                  #等待时间5s
splashimage                                #背景图 .xpm格式 gz压缩
hd0,0                                      #相当于sda1，grub下第一块硬盘第一个分区
hiddenmenu
title后
root(hd0,0)                                #以sda1为根
kernel /vmlinuz-2.6.18-128.e15             根下
root=dev/VolGroup00/LogVol00               切换 3次
rhgb                                       #引导图形界面 fdisk -l：boot下*代表可引导
quiet
initrd
```

3. kernel —— 内核 —— vmlinuz-2.6.18-128.e15 /boot/

系统镜像 —— initrd-2.6.18-128.e15.img

4. init —— 系统初始化 —— 运行级别

例1: cd /boot

mkdir /boot/initrd

gunzip < initrd-2.6.18-128.el5.img > initrd/initrd.img

#解压

cpio -iv < initrd/initrd.img

#释放

ls /boot/initrd

#bin dev etc init lib proc sbin sys sysroot

例2: dd </dev/sda > /tmp/mbr bs=512 count=1

#从设定的文件中读内容，二进制读写. cp

bs:一次读多少，count:读几次，bs=512是mbr

cat /tmp/mbr

hexdump -C /tmp/mbr

#乱码

#将二进制文件转换为十六进制输出，-C将十六进制转换为ASCII码

No.

Date

三. init 运行级别

0 关机

1 单用户模式，所有服务都不启动

2 缺省模式

3 完全模式，只运行字符界面

4 未设置

5 3+桌面

6 重启

※ /etc/inittab

```
id:5:initdefault:                          # initdefault 默认运行级别
si::sysinit:/etc/rc.d/rc.sysinit           # 没写级别是在任何级别都执行
                                           最后一个冒号后面是要执行程序的路径

/etc/rc.d/rc.sysinit                       # 初始化计算机环境，并挂载磁盘
                                           内的可读可写状态。
/etc/rc.d/*                                # 不同运行级别的服务目录
    rc5.d                                  ./rc 3

/etc/rc5.d                                 # K开头stop，S开头start

/etc/init.d/syslog stop                    # 关闭内核日志记录器。
service syslog start                       # 启动内核日志记录器
```

手动引导

e2label /dev/VolGroup00/LogVol00 hello

c↙

root (hd0,0)

kernel /vmlinuz-2.6.18-128.el5 ro root=LABEL=hello init=/bin/bash

initrd /initrd-2.6.18-128.el5.img

boot ↙

/etc/rc.d/rc.sysinit　　#初始化计算机环境，并获取磁盘可读可写

/etc/rc.d/rc 1　　#以运行级别1运行

startx　　#进入桌面

修改grub密码

vim /boot/grub/grub.conf　　#明文

在title上面加password 123456

grub-md5-crypt ↙　　#加密

123456

得散列值

vim /boot/grub/grub.conf

password --md5 散列值

/etc/issue　　#修改登录提示

No.

存储管理

Date 2009 · 8 · 12

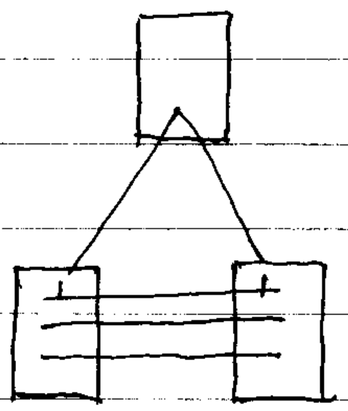
文件系统

分　区 ⟶ 空间　　LVM 逻辑卷管理

磁　盘 ⟶ 速度、冗余　　RAID 磁盘阵列

一. RAID（级别 0 1 5）

1. 0 速度 并行写入 最少2块磁盘

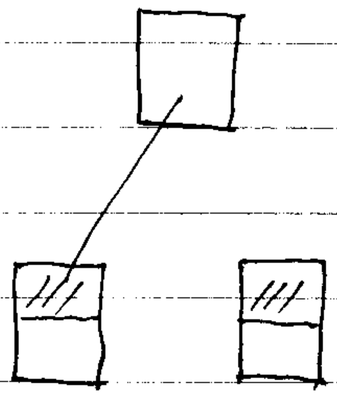


条带为基本物理单元 chunk 默认 64

条带数　64 (n-1)　n是硬盘数

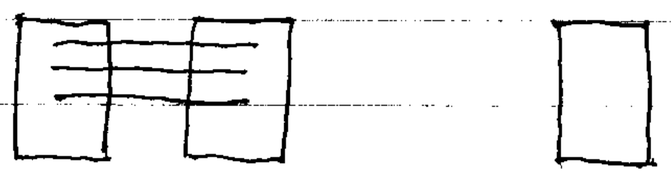
速度　m (n-1)　m是每块硬盘的速度

2. 1. 冗余 镜像 最少2块磁盘



实时同步

3. 5 速度、冗余， 最少3块磁盘 ，只能提供1块磁盘的冗余



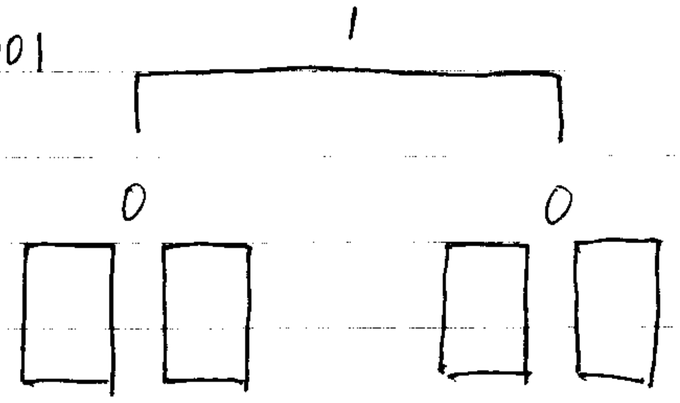
条带化

0101　1100

按位做奇偶校验，异或算法，相同为0不同为1

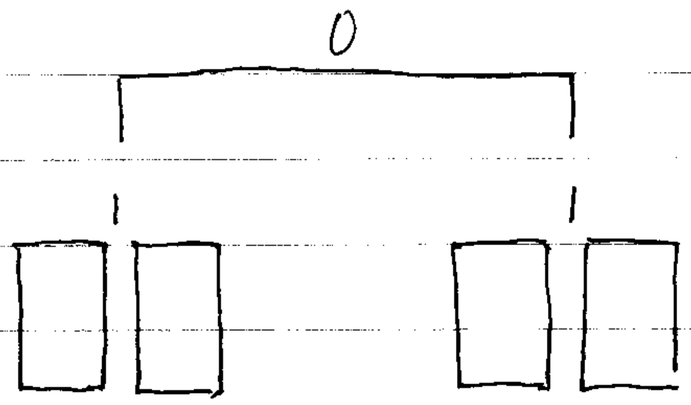
1001

4. RAID01



一侧的两台可以一起坏，不可以两侧各坏一台

5. RAID10



两侧可以各坏一台，不可以一侧的两台一起坏

例：以sda8、sda9、sda10三个分区模拟RAID 5，每个分区1G大小

```
rpm -q  mdadm                          #查此命令的安装包
mdadm -C /dev/md0 /dev/sda{8,9,10}     #创建第一个RAID设备
      -l 5                             指定级别
      -n 3                             指定设备数
      -c 32                            每块做多条带，级别0,5用，默认64
```

```
cat /proc/mdstat                       #查看当前RAID状态
```

```
mke2fs -j /dev/md0                     #格式化
mount /dev/md0 /mnt                    #挂载
mdadm -D /dev/md0                      #查看
mdadm -Ds                              #检测
mdadm -Ds > /etc/mdadm.conf            #开机自动识别，配置文件
```

No.

Date

编写脚本后台运行使每0.1秒创建一个文件

vim test.sh

```
count=0
while :
do
dd if=/dev/zero of=/mnt/file.$count bs=$[RANDOM%900+100] count=1000
usleep 100000      count=$[$count+1]
done
```

```
#: if=input file      of=output file      %除法取余,
echo 产[$RANDOM%900+100] 三位数,随机
```

```
chmod +x test.sh                    #加执行权限
./test.sh &> /dev/null &            #后台运行
```

```
df -h                               #包含1G不显示,
```

```
mdadm /dev/md0 -f /dev/sda10        #人为损坏
```

```
mdadm /dev/md0 -r /dev/sda10        #热删除
```

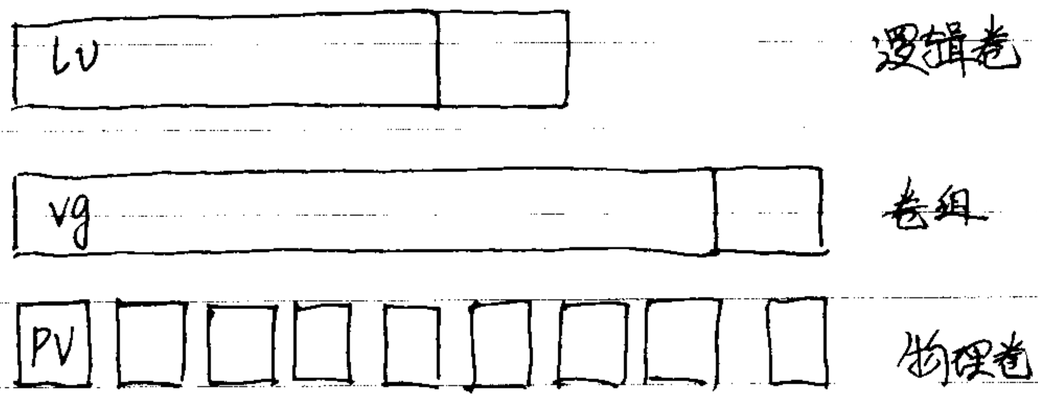
```
mdadm /dev/md0 -a /dev/sda11        #添加一个分区
```

```
watch cat /proc/mdstat              #监控
```

```
dd if=/dev/zero of=/mnt/file ↵      #测速
```

```
mdadm -S /dev/md0                   #停掉 RAID
mdadm -A /dev/md0                   #启动,装载
```

二. LVM 逻辑卷管理



例：使用分区 sda{8,9,10} 创建逻辑卷2G，并格 lv，vg.

```
pvcreat  /dev/sda{8,9,10}                  #创建pv
vgcreat  vgtest  /dev/sda{8,9,10}          #创建 vg，vgtest —vg名称
lvcreat  -n  lvtest  -L  2G  vgtest        #划分lv，-n：lv名称，-L：指定空间
/dev/vgtest/lvtest                         #lv设备文件          一个：支错多入加磁盘数

mke2fs -j  /dev/vgtest/lvtest              #格式化
mount  /dev/vgtest/lvtest  /mnt            #挂载

lvdisplay                                  #查看lv信息，逻辑卷
vgdisplay                                  #    vg      卷组
pvdisplay                                  #    pv      物理卷

lvextend  /dev/vgtest/lvtest  -L  +500  M  #扩容LV，-L后可跟总空间大小，也可新加空间
                                                文件系统支持扩容
df -h                                      #2G
resize2fs  /dev/vgtest/lvtest              #另格式化新加空间
df -h                                      #2.5G

pvcreat  /dev/sda11
vgextend  vgtest  /dev/sda11               #扩容vg
```

No.

Date

```
umount    /mnt
lvremove  /dev/vgtest/lvtest
vgremove  /dev/vgtest
pvremove  /dev/sda{8,9,11}
```

系统管理——计划任务

```
service atd restart          #重启临时计划任务
service crond restart        #重启周期计划任务
```

一、临时任务 at

```
at now ↵                     #现在立即执行
at> ls -l ↵                  编辑命令
at> ctrl+d                   退出
mail ↵                       #查看邮件
at noon                      #正午12点
at teatime                   #下午16:00点
at 13:44                     #未来24小时内
at 1:44 pm
at 20:00 12302010            #在2010年12月30日20点执行

atq =at -l                   #查看计划任务队列
at -c 3                      #查看编号为3的计划任务详细信息

atrm 7                       #删除任务
at -d 3
at -r 5

ezfsck -fy /dev/sda1         #对文件系统进行强制检测并修复,先卸载
```

二、周期任务 cron

crontab -u zorro -e ↵　　#定义周期性，不加-u默认当前用户

	0~59	0~23	1~31	1~12	0~6	
五段	分钟	小时	日	月	周	命令 写绝对路径，tab分隔
	*	*	*	*	*	ls -l　#每月每周每天每小时每分钟执行1次
	01	*	*	*	*	#每小时 第01分钟
	30	0	*	*	*	#每天0:30
	30	01	*	*	0	#每周日01:30
	40	11	3	*	0	#每月3号且是周日11:40
	1,31					#每30分
	*/5					#每隔5分
	1-5					#第1.2.3.4.5分钟

crontab -l　　#查看周期任务

crontab -r　　#清空周期任务

vim /etc/crontab

九段	02	4	*	*	*	root	run-parts /etc/cron.daily

#run-parts执行后面目录下可执行的文件

vim /etc/cron.daily/mlocate.cron　　#更新数据库

/usr/bin/updatedb -f

servie anacron start　防止遗漏

vim /etc/anacrontab　　# anacron配置文件，开机执行

四段	1	65	cron.daily	run-parts	/etc/cron.daily

检查一天内是否执行，如未执行延期65分钟执行

vim /var/spool/anacron/cron.daily　　#上一次执行的时间

20090812

例 vim ls.sh 添加到 /etc/cron.daily/　　#自定义系统计划任务

chmod +x ls.sh

No.

Date

日志.

service syslog restart

dmesg　　#查看即时日志.

/var/log/messages

/etc/syslog.conf　　# man 5 syslog.conf 查看对象.级别等

*.info;mail.none;news.none;authpriv.none;cron.none　　/var/log/messages

所有对象.级别　不记录　　日志存放目录文件

#对象*不包含mark. mark跟踪syslog,每隔一段时间向syslog做标记,表示syslog正常

#级别info及以上,=info只这一种级别

#mail,news,authpriv,cron单独记录

-/var/log/maillog　　#默认日志记录是同步,不经过缓存

-表示经过缓存,mail日志大同步速度慢

root　　#报告给root用户终端

*　　#报告给所有用户终端

/dev/tty11　　#用一个空闲终端监控日志

@192.168.1.254　　#远程日志监控

254: ps ax | grep syslog

vim /etc/sysconfig/syslog　　#设置远程日志接收 -r

-m 0 关闭mark, -x不解析

#tabels -F　　#关闭防火墙

netstat -anu　　#查看端口,远程日志记录514

local0-7　　#自定义对象

echo 111 | logger

tail -f /var/log/messages

vim /etc/syslog.conf　　#添加local1.*　/var/log/local1.log

echo 111 | logger -p local1.info

tail -f /var/log/local1.log

日志轮转

```
/etc/cron.daily/logrotate                    #开机执行，检查logrotate是否执行

/etc/logrotate.conf                          #查日志是否到期
    weekly                                   #每周记录
    rotate 4                                 #保存之前4周日志，
    creat                                    #创建新日志 1 ，上一次1改为2
    include /etc/logrotate.d

/var/log/wtmp {
    monthly                                  #每月记录
    minisize 1M                              #最小 1M
    create 0664 root utmp                    #权限，属主、属组
    rotate 1                                 #保存之前1个月的日志
}
```

No.

网络管理

Date 2009 · 8 · 13

```
arp -n                                  #显示arp缓存
arp -s 192.168.1.1 00:11:22:33:44:55    #绑定mac地址

tcpdump -i eth0 -n icmp                 #抓包，ping命令用icmp协议
```

Linux专用命令

```
ip link show                            #显示所有网卡，包括down
ip link sh eth0                          =ip link list

ip link set eth0 address 00:11:22:33:44:55   #修改mac地址

ip neigh show                           #查看arp缓存
ip ne add 192.168.1.1 lladdr 00:11:22:33:44:55  #给ip绑定mac

ip address show                         #查看ip信息
ip addr show dev eth0                   #查看eth0 ip信息

ip addr add dev eth0 10.0.0.1/8         #添加ip地址，ifconfig看不到
ip addr del dev eth0 10.0.0.1/8         #删除ip地址，换一个ip要先删除

ip route show                           #查看路由

ip ro add default dev eth0 via 192.168.1.1   #添加默认网关
ip ro add 10.0.0.0/8 dev eth0 via 192.168.1.1  #发往一个网段
ip ro add 10.0.0.3/32 dev eth0 via 192.168.1.1  #发往主机
```

内核驱动

Date 2009 8 14

lsmod #查看当前内核中已加载的驱动

lsmod | grep 8139

8139too

8139cp

mii 8139too, 8139cp, r8169

rmmod 8139too #"删除", 断开加载

ifconfig eth0 192.168.1.254/24 #添加internet内核自动加载驱动

mii-tool #内核检查自动加载驱动

modprobe 8139too #手动加载驱动,自动解决依存关系,

modprobe -r 8139too #"删除"

insmod /lib/modules/2.6.18-128.el5/kernel/drivers/net/8139too.ko

#手动加载驱动不解决依存关系,

modinfo 8139too #查看一个模块的信息.

depends mii depends: 给谁提供依存关系,

license GPL GPL开源, BSD

parm parm: 参数

modinfo e1000

/lib/modules/2.6.18-128.el5/modules.dep #依存关系文件

depmod #更新依存关系文件

/lib/modules/2.6.18-128.el5/kernel/ #内核驱动

/lib/modules/2.6.18-128.el5/kernel/drivers/ #设备驱动

/sound

/lib/modules/2.6.18-128.el5/kernel/net/netfilter #防火墙驱动

/ipv4

www.kernel.org #内核源码下载

编译内核

44M

tar jxf linux-2.6.25.tar.bz2 -C /usr/local/src

cd /usr/local/src/linux-2.6.25

du -sh 389M

ls

arch #支持哪些CPU架构

block #块接口层驱动

crypto #加密算法的实现

Documentation #

drivers #设备驱动源码

init #初始化

ipc #进程间通信

kernel #内核本身支持的信息.

lib #内核库文件

mm #内存管理

net #网络协议

samples #

scripts #脚本

security #安全

sound #声音

usr #另外的引导程序

virt #内核的虚拟化

1. make menuconfig #配置内核的菜单

* 编译到内核里.

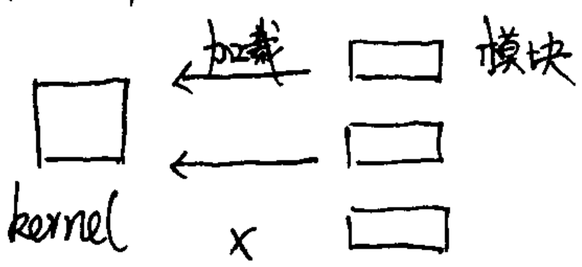
M 模块在/lib/

保存为.config文件

2. make && make modules_install && make install

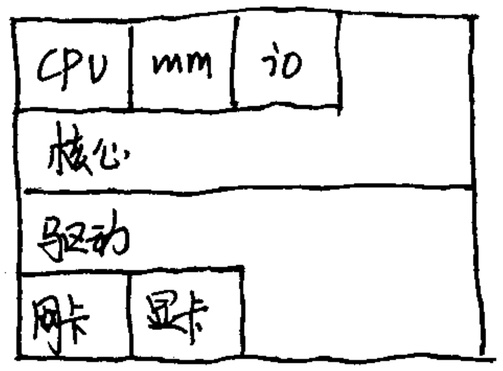
编译 && 安装模块 && 安装内核

/boot/vmlinuz-2.6.18-128.el5　　#内核，1.8M.压缩镜像

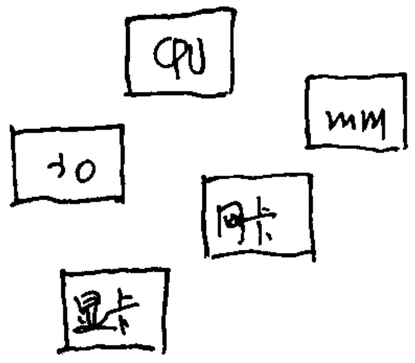


#编译到内核里或内核外

巨内核(单体式内核)传统Unix



微内核



不同功能拆开实现
理论上更好

mknod 123 b 8 0

#手动建立一个设备文件(文件名和路径不重要，设备号最重要)

123：文件名

b：块设备

8：主设备号，标识使用何种驱动

0：从设备号，第一块硬盘

ls -l 123　　#相当于 ls -l /dev/sda

rm 123　　#删除设备文件

mount /dev/cdrom /mnt

cp -rf /mnt/* /var/ftp/pub/RHEL5U3

find /var/ftp/pub/RHEL5U3 -name kernel-doc*

rpm -ivh kernel-doc-2.6.18-128.el5.noarch.rpm　　内核的说明信息包

cd /usr/share/doc/kernel-doc-2.6.18/Documentation

less devices.txt

设备名称	主设备号	设备类型	从设备号	
IDE	3	block	0 = /dev/hda	0：整个硬盘
			64 = /dev /hdb	1-63：分区
			⋮ ⋮	
			192 d	
SCSI	8	block	0 = /dev/sda	0：整个硬盘
			16 = /dev/sdb	1—15：分区
			⋮ ⋮	
			240 p	
SATA	160	block	0 =/dev/carmel 0	0：整个硬盘
			32= /dev/carmel 1	1—31：分区
			⋮ ⋮	/dev/carmel0p1
			24 7	表示第1块硬盘的第1个分区

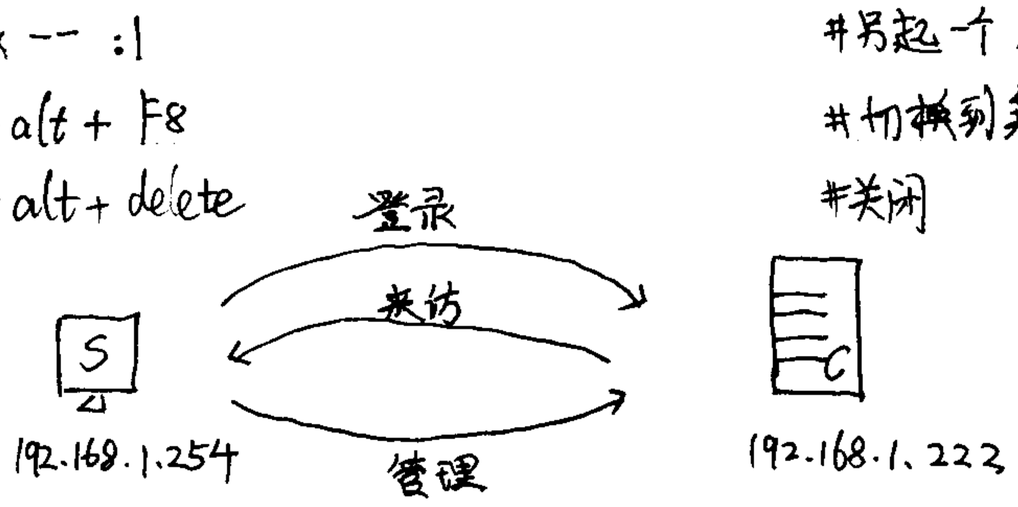
X-window

1. C/S模式，客户端服务器基于X协议，X11第11版

startx ⟶ ①X window server ，监听127.0.0.1
②X client ，自己的客户端输出到自己的服务端

```
X ↵                          #打开服务端
xinit                        #打开服务端,并打开自己客户端一个shell
xterm                        #X终端
xclock                       #时钟
firefox                      #浏览器
startx                       #打开服务端并打开DE桌面环境
gnome-session                #gnome桌面
startkde                     #kde桌面
startx -- :1                 #另起一个x-window
ctrl+alt+F8                  #切换到第二个x-window
ctrl+alt+delete              #关闭
```

2. 远程管理




```
xinit                                  #打开服务端
xhost + 192.168.1.222                  #指定来访IP,不写IP任何人可访问
ssh 192.168.1.222                      #登录客户端
export DISPLAY="192.168.1.254:0"       #设置环境变量,0.第1个X-window 服务端的
gnome-session || startkde              #打开gnome或kde桌面环境
```

3. kernel 模块 /lib/modules/2.6.18-128.el5/kernel/drivers

桌面

/usr/lib/xorg/modules/drivers/intel_drv.so

vim /etc/X11/xorg.conf
system-config-display

#更改桌面环境配置文件
#命令更改，更改后会将原来的配置文件 xorg.conf
备份为 xorg.conf.backup

vim /usr/bin/startx #脚本
userclientrc = $HOME/.xinitrc
sysclientrc = /etc/X11/xinit/xinitrc

startx → X ~/.xinitrc
✓ /etc/X11/xinit/xinitrc
↳ X ~/.xclients
✓ /etc/X11/xinit/xclients
↳
/etc/sysconfig/desktop
↳ GNOME 默认
KDE

vim /etc/sysconfig/desktop
空文件
写入 DESKTOP="KDE"
startx启动 KDE桌面

ULE-116

服务

2009 8 17

1. 服务定义与控制
2. 地址相关服务
   DHCP
   DNS
3. 文件相关服务
   WEB squid
   FTP
   NFS
4. 电子邮件系统
   send mail
   postfix
5. 网络信息系统
   NIS
6. 远程管理用户
   SSH
7. 安全
   PAM
   TCP- WRADER
   IPTABLES

服务相关命令

standalone配置文件存放于/etc目录下
xinetd配置文件存放于/etc/xinetd.d下
xinetd服务本身的配置文件为/etc/xinetd.conf

/etc/services 定义了端口和服务

/etc/ 独立服务 httpd

/etc/xinetd.d/ 不常用的服务，监听相应的每个服务的端口，有请求时再启动相应的服务

```
chkconfig krb5-telnet on                    #打开监听服务
service xinetd restart

/etc/rc.d/init.d/httpd restart

chkconfig --list                            #脚本在各运行级别启动否
chkconfig --list httpd
chkconfig --level 1 httpd on|off|reset

ntsysv                                      #默认在当前运行级别上开启或关闭服务
ntsysv --level 35                           #在运行级别3,5设置
```

/etc/rc.d/rc3.d/符号连接

```
K                                           #开机不启动
S12                                         #开机启动，字母后面的数字是启动次序

chkconfig --level 3 auditd off
ls -l |grep auditd                          #S开头变K开头

          3    5
sshd      on   off
当前在3级别，init 5                          #sshd变K开头
```

No

Date

dhcp

1. 端口 UDP 67.68， dhcpd

2. 安装包 dhcp-3.0.5-18.el5

3. 配置文件 /etc/dhcpd.conf

4. 修改配置文件

```
cp  /usr/share/doc/dhcp-3.0.5/dhcpd.conf.sample     /etc/dhcpd.conf
vim  /etc/dhcpd.conf                                #拷贝模板，修改
```

```
subnet  192.168.1.0     netmask 255.255.255.0       #网段
option  routers                                     #网关
option  subnet-mask                                 #子网掩码
option  domain-name-servers                         # dhcp server，比行前2行#注释掉
range   192.168.1.89    192.168.1.253               #地址池
default-lease-time  21600                           #默认租约6小时
max-lease-time  43200                               #最大到期时间12小时
hardware  ethernet                                  #绑定IP
fixed-address
```

5. service dhcpd restart

6. 日志，

```
/var/lib/dhcpd/dhcpd.leases                         #dhcp服务端日志，
    starts                                          #更新租约时间
    ends                                            #最大到期时间
    renew                                           #更新
    rebind                                          #重试更新
    expire                                          #过期时间

/var/lib/dhclient/dhclient-eth0.leases              #dhcp客户端日志，
```

dns

1. 53端口， named ， UDP

2、安装包

bind-9.3.4-10.P1.el5 #主程序
bind-utils-9.3.4-10.P1.el5 #工具套件
bind-libs- 9.3.4-10.P1.el5 #库文件
bind-chroot -9.3.4-10.P1.el5 #安全插件
caching-nameserver #模板文件

3. 配置文件

/var/named/chroot/etc/named #配置文件
/var/named/chroot/var/named/com.db #数据文件

/var/named/chroot/etc/named.caching-nameserver.conf
/var/named/chroot/etc/named.rfc1912.zones #配置文件模板

4. 解析方式

递归
迭代

5. FQDN，完整的主机名称解析，如 www.163.com

6. 每个DNS服务器上有13个根域服务器

/var/named/chroot/var/named/named.ca

7. SOA记录：起始授权，为存储在区域中的信息指明授权机构的起点并标始记录
nameserver NS记录：名称服务器，用来指定该域名由哪个DNS服务器进行解析
address A记录：指定主机名或域名对应的IP地址记录

TTL ：缓存保存时间

正向解析

```
mv  /var/named/chroot/etc/  named.caching-nameserver.conf  /root/
                            named.rfc1912.zones
                                                  #将模板移除,

※ vim  /var/named/chroot/etc/named.conf  #修改配置文件

options {                                 #选项        声明数据文件所在文件夹
        directory   "/var/named";         #相对路径 /var/named/chroot/var/named
        dump-file   "/var/named/data/cache_dump.db";
};
logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
zone "88.com" IN {
                type master;
                file "88.com.db";
};
:wq

※ vim  /var/named/chroot/var/named/88.com.db  #修改数据文件

$TTL  86400
88.com.          IN   SOA   dns.88.com.   root\@88.com.  (2009081701  3H  15M  1W
                                                                          1D)
88.com.          IN   NS    dns.88.com.
dns.88.com.      IN   A     192.168.1.88
www.88.com.      IN   A     192.168.1.88
mail.88.com.     IN   A     192.168.1.88
:wq
```

※ chmod 640 88.com.db
chown root:named 88.com.dfp

service named start
tail -f /var/log/messages #监控日志

客户端查询

vim /etc/resolv.conf

nameserver 192.168.1.88

host www.88.com

iptables -L
iptables -F #关掉防火墙

getenforce

子域授权　　父域 192.168.1.254　　子域 192.168.1.88

```
#vim /var/named/chroot/etc/named.conf   #修改配置文件

zone "com" IN {
        type master;
        file "com.db";
};
```

```
#vim /var/named/chroot/var/named/com.db   #修改数据文件

$TTL 86400
com.        IN  SOA  dns.com.  root@com. (2009081701 3H 15M 1W 1D)
com.        IN  NS   dns.com.
dns.com.    IN  A    192.168.1.254

88.com.     IN  NS   dns.88.com.
dns.88.com. IN  A    192.168.1.88
:wq
```

```
#chmod 640 com.db
chown root:named com.db

service named start
tail -f /var/log/messages
```

客户端查询

```
vim /etc/resolv.conf
nameserver 192.168.1.254

host www.88.com
```

主从备份

主服务器　192.168.1.88

vim /var/named/chroot/etc/named.conf　#修改配置文件

```
zone "88.com" IN {
        type master;
        file "88.com.db";
        allow-transfer {192.168.1.100; };
};
:wq
service named restart
```

从服务器　192.168.1.100

```
vim /var/named/chroot/etc/named.conf

zone "88.com" IN {
        type slave;
        file "slaves/88.com.db";    相对路径
        masters {192.168.1.88; };
};
:wq
service named restart
```

/var/named/chroot/var/named/slaves/88.com.db

```
$ORIGIN .                          #补.
$TTL 86400      ; 1 day            #;注释，从服务器的TTL
88.com          IN SOA dns.88.com. root\@88.com. (
                                   2009081702 ; 序列号
```

10800 ; 到服务器上更新数据的时间，3 hours
900 ; 重试更新，15 minutes
604800 ; 不再更新，week
86400 ; 1 Day
)

$ORIGIN 88.com.
dns A 192.168.1.88

纯缓存 DNS服务器，没有 zone

```
vim /var/named/chroot/etc/named.conf      #修改配置文件

options {
        forward only;                     #只，转发
        forwarders { 192.168.1.88; };
        directory "/var/named";
        dump-file "/var/named/data/cache_dump.db";
};
:wq
service named restart

cd /var/named/chroot/var/named/data/
rndc dumpdb -cache                        #产生、更新文件 cache_dump.db
cache_dump.db
less cache_dump.db                        #查看缓存
```

TTL 倒计时 dig www.88.com
rndc dumpdb -cache
less cache_dump.db

2009 8 18

host www.sohu.com

www.sohu.com is an alias for d7.a.sohu.com.
d7.a.sohu.com is an alias for pgcnctct07.a.sohu.com.

pgcnctct07.a.sohu.com has address 61.135.179.155
160
184
190
61.135.133.37
38
88
89

DNS负载均衡

```
$TTL 86400
uplooking.com.        IN    SOA    dns.uplooking.com.  root\@uplooking.com.( )
uplooking.com.        IN    NS     dns.uplooking.com.
dns.uplooking.com.    IN    A      192.168.1.254

www.uplooking.com.    0     IN     A     124.42.124.60    #此缓存服务器对这些记录不缓存
www.uplooking.com.    0     IN     A     61
www.uplooking.com.    0     IN     A     62
```

— ping www.uplooking.com
62
61
60

uplooking.com.	IN	MX	10	mail1.uplooking.com.
uplooking.com.	IN	MX	20	mail2.uplooking.com.
uplooking.com.	IN	MX	30	mail3.uplooking.com.
mail1.uplooking.com.	IN	A		124.42.124.66
mail2.uplooking.com.	IN	A		67
mail3.uplooking.com.	IN	A		68

10.20.30为优先级，10的优先级最高，MX为邮件记录

web.uplooking.com.	IN	CNAME	www.uplooking.com.
www.uplooking.com.	IN	A	124.42.124.60

CNAME为别名记录

— host web.uplooking.com.
web.uplooking.com is an alias for www.uplooking.com.
www.uplooking.com has address 124.42.124.60

— host uplooking.com
uplooking.com mail is handled by 10 mail1.uplooking.com
20 mail2
30 mail3

— host mail1.uplooking.com
mail1.uplooking.com has address 124.42.124.66

apache
1.端口 80，可修改， httpd tcp

2. 安装包
httpd-2.2.3-22.el5 #主程序
httpd-del-2.2.3-22.el5 #开发工具
httpd-manual-2.2.3-22.el5 #手册

3. 配置文件
/etc/httpd/conf/httpd.conf #主配置文件
/conf.d/vhost.conf #独立功能配置文件 包含于主配置文件中.
/logs/access_log #日志.
/modules/libphp5.so #apache的模块，1个模块1个功能
/var/www/html/index.html # DocumentRoot，索引文件

4. service httpd restart
/etc/rc.d/init.d/httpd restart

5. 别名
vim /etc/httpd/conf.d/alias.conf
alias "/wxs" "/etc/fstab"
http://192.168.1.88/wxs

6. 虚拟主机 #配置虚拟主机后 /var/www/html 无文件
vim /etc/httpd/conf.d/vhost.conf
NameVirtualHost *:80
<VirtualHost *:80>
DocumentRoot /var/www/uplookng
ServerName www.uplookng.com
</VirtualHost>
httpd -S #查看默认虚拟主机，http://192.168.1.88

No.

Date

```
vim /var/named/chroot/var/named/com.db
$TTL 86400
com.        IN  SOA  dns.com.root\@com.  (20090818| 3H 15M 1W 1D)
com.        IN  NS   dns.com.
dns.com.    IN  A    192.168.1.88
www.sohu.com.   IN  A   192.168.1.88
www.sina.com.   IN  A   192.168.1.88
www.163.com.    IN  A   192.168.1.88
:wq
chmod  640  com.db
chown  root.named  com.db
/etc/rc.d/init.d/named  restart
host  www.sohu.com                          #确定DNS配置正确


cd  /var/www
mkdir  sohu  sina  163
cd sohu
vim index.html     Welcome to sohu.com      #验证 squid
cd sina
vim index.html     Welcome to sina.com      #验证用户密码访问
cd 163
mkdir dir                                   #验证索引，访问控制
cd dir
touch  aa  bb  cc  dd

cd  /etc/httpd
htpasswd -c password blues  123
htpasswd password zorro  123
```

```
vim  /etc/httpd/conf.d/vhost.conf

NameVirtualHost  *:80

<VirtualHost  *:80>                                    #虚拟主机
      DocumentRoot     /var/www/sohu
      ServerName       www.sohu.com
</VirtualHost>

<VirtualHost  *:80>
      DocumentRoot     /var/www/sina
      ServerName       www.sina.com
          <Directory   "/var/www/sina">      #目录容器
              authname   "Welcome..."
              authtype   basic
              authuserfile   "/etc/httpd/password"
              require  valid-user
          </Directory>
</VirtualHost>

<VirtualHost  *:80>
      DocumentRoot     /var/www/163
      ServerName       www.163.com
          <Directory   "/var/www/163/dir">
              Options  Indexes                 #无index.htm[时建立索引]
                                                http://www.163.com/dir

              order allow,deny
              allow from all
              deny from 192.168.1.245
          </Directory>
</VirtualHost>
```

squid 缓存代理

1. 端口 3128 , squid

2. 安装包 squid-2.6.STABLE 21-3.el5.i386.rpm

3. 配置文件 /etc/squid/squid.conf

4. 修改配置文件

```
cp squid.conf squid.conf.bak
grep -v ^# squid.conf | grep -v ^$          #删去注释和空行
vim /etc/squid/squid.conf

http_port 3128                              #端口
cache_mem 8 MB                              #代理服务器占内存大小
cache_swap_low 90                           #
cache_swap_high 95                          #缓存中的文件大于95%开始删除,
maximum_object_size 4096kB                  #大于4096kB的文件不缓存
cache_dir nfs /var/spool/squid 100 16 256
visible_hostname server.com                 #代理服务器域名
acl all src 0.0.0.0/0.0.0.0
http_access allow all
```

5. service squid start

6. 浏览器 — Edit — Preferences — Advanced — Network — Settings — Manual proxy configration:
   HTTP Proxy : 192.168.1.88 Port : 3128

7. http://www.sohu.com Welcome to sohu.com

```
echo nihao > /var/www/sohu/index.html
elinks www.sohu.com                         nihao
http://www.sohu.com                         Welcome to sohu.com
F5刷新                                      nihao
```

ftp 2009 8 19

1. 端口21认证端口，主动模式 端口20传输，被动模式 大于1024的任意端口传输，vsftpd

2. 安装包 vsftpd-2.0.5-12.el5

3. 配置文件

```
/etc/vsftpd/vsftpd.conf                    #配置文件
/etc/vsftpd/ftpusers                       #系统黑名单
           /user_list
/etc/vsftpd/chroot_list                    #手动创建黑名单
/var/ftp                                   #匿名用户登录默认在/var/ftp/pub下
```

4. 修改配置文件

vim /etc/vsftpd/vsftpd.conf

① anonymous：匿名用户

```
anonymous_enable=YES                       #允许匿名用户登录
anon_upload_enable=YES                     #允许匿名用户上传
anon_mkdir_write_enable=YES                #允许匿名用户新建目录
anon_other_write_enable=YES                #允许匿名用户删除
anon_max_rate=30000                        #匿名用户最大下载速率，单位字节
```

② local user：本地帐户

```
local_enable=YES                           #允许本地帐户登录
local_umask=022
chroot_local_user=YES
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/chroot_list
userlist_enable=YES
pam_service_name=vsftpd
local_max_rate=50000                       #本地用户最大下载速率
```

No

Date

③ log：日志.

dual_log_enable=YES　　#双日志.
xferlog_enable=YES　　#上传下载日志.
dirmessage_enable=YES　　#目录信息.
xferlog_std_format=YES

④ other：其它

write_enable=YES　　#全局写
connect_from_port_20=YES　　#端口模式即主动模式
max_per_ip=3　　#每个IP最大线程数
max_client=10　　#最大客户端连接数
listen=YES
tcp_wrappers=YES

5. service vsftpd restart

6. 客户端

ftp

lftp 192.168.1.254　　#有端匿名登录
lftp -u username 192.168.1.254,passwd

gftp　　#图形界面
192.168.1.254　　21　　anonymous

7. 注意事项

无法在pub下上传文件，可在 pub/anon 权限为777的目录下上传

2049端口

nfs 网络文件系统

1. 无固定端口，portmap 在 CS之间协定一个端口，portmap的端口为 111

2. 安装包

nfs-utils-1.0.9-40.el5

nfs-utils-lib-1.0.8-7.2.z2

3. 配置文件

/etc/exports

4. 修改配置文件

vim /etc/exports

/tmp *(rw,async) # 读写 异步传输

/var/ftp/pub 192.168.1.0/24(rw,sync) 192.168.0.0/255.255.0.0(ro)

#共享给多个子网

5. service portmap start

service nfs start

6. showmount -e 192.168.1.254 #查看共享

mount 192.168.1.254:/var/ftp/pub /mnt #挂载

7. 注意事项

无法在pub下创建文件，可在pub/anon(777)目录下创建

cd anon

touch file1 允许

ll file1

-rw-r--r-- 1 nfsnobody nfsnobody

root帐户创建身份改为nfsnobody，引到普通帐户创建，用户名不变.

ll -n 显示 uid，gid 而不显示用户名.

autofs 自动挂载

1. 端口

2. 安装包

3. 配置文件

/etc/auto.master

/etc/auto.misc

4 修改配置文件

vim /etc/auto.master

/misc/ /etc/auto.misc # /misc/ 挂载点，在/misc下 cd 即挂载 /etc/auto.misc定义挂载目录

vim /etc/auto.misc

nfs -ro,soft,intr 192.168.1.254:/var/ftp/pub # 在/misc/下 cd nfs 即挂载192.168.1.254 /var/ftp/pub目录到 /misc/

5. service autofs restart

6. cd /misc/ ls

ls 空 RHEL5U3

cd nfs

7. vim /etc/sysconfig/autofs

TIMEOUT=300 # 300秒不使用，自动卸载

samba

1. smb

2. 安装包

samba-3.0.33-3.7.el5

samba-client-3.0.33-3.7.el5

samba-common-3.0.33-3.7.el5

samba-swat.i386 0:3.0.33-3.7.el5

3. 配置文件

/etc/samba/smb.conf

4. 修改配置文件

vim /etc/samba/smb.conf

```
[global]
        workgroup = WORKGROUP
        server string = File server
        hosts allow = 127.  192.168.1.          #允许访问的网段
        log file = /var/log/samba/%m.log
        max log size = 50
        security = user                         #user系统帐户机制,share匿名用户机制
        passdb backend = tdbsam

[homes]
        comment = Home Directories
        browseable = no
        writable = yes
```

No.

Date

[Public]

comment = Public Stuff
path = /var/ftp/pub　　#共享目录路径
public = yes
browseable = yes　　#隐藏共享目录

5. service smb restart

6. 登录

smbclient -L //192.168.1.254　　#查看共享目录
smbclient //192.168.1.254/public　　#匿名登录,不用输密码

smbpasswd -a up　　#添加samba用户,up系统中确实存在的用户名
-x　　#删除,
smbclient //192.168.1.254/up -U up　　#验证帐户登录.

7. 挂载

mount.cifs //192.168.1.254/public /mnt　　#匿名用户挂载
mount.cifs //192.168.1.254/public /mnt -o username=up,password=123,
tocharset=cp936
#验证帐户挂载,多个参数用,隔开.
tocharset 指定字符集

8. 不能在pub下上传,可在pub/anon(777)目录上传.
put /root/install.log huhy
上传后的文件名

9. chkconfig swat on
   service xinetd restart

   netstat -anpt | grep 901　　#检查901端口是否开启

   http://127.0.0.1:901/

No.

Date 2009 8 21

sendmail 25端口

1.安装包

sendmail-8.13.8-2.el5 #主程序
sendmail-cf-8.13.8-2.el5 #配置文件模板
sendmail-doc-8.13.8-2.el5 #手册
sendmail-devel-8.13.8-2.el5 #开发工具

2.配置文件

/etc/mail/sendmail.mc #配置文件
/etc/mail/sendmail.cf

/var/log/maillog #日志
/var/spool/mqueue #队列
/var/spool/user

/etc/aliases #群发
/etc/mail /virtusertable #虚拟账户转发
/etc/mail/virtusertable.db
~/.forward #转发

/etc/mail/access #访问控制
/etc/mail/access.db

3.

MUA：邮件客户端代理，如foxmail，outlook
将用户的邮件通过SMTP协议发送至MTA
MTA：邮件传输代理，如sendmail，postfix，exchange
将来自MUA的邮件通过SMTP转发
MDA：常和MUA合二为一，如foxmail，outlook
通过POP3，IMAP协议从MTA取得邮件并传送至邮件接收者的邮箱
SMTP：简单邮件传输协议，端口25
POP3：邮件协议，从邮件服务器下载电子邮件到本地，端口110
IMAP：通过客户端直接对邮件服务器上的邮件进行操作，端口143
dovecot：POP3，IMAP

搭建邮件系统

一. DNS

父域：

1. 建立com域

vim /var/named/chroot/etc/named.conf

```
zone "com" IN {
                type master;
                file "com.db";
};
```

2. 子域授权

vim /var/named/chroot/var/named/com.db

```
$TTL 86400
com.            IN  SOA  dns.com.  root@com. (2009082 3H 15M 1W 1D)
com.            IN  NS   dns.com.
dns.com.        IN  A    192.168.1.254

uplooking.com.  IN  MX   10  server.uplooking.com.
server.uplooking.com.  IN  A      192.168.1.254

1.com.          IN  NS   dns.1.com.
dns.1.com.      IN  A    192.168.1.1
2.com.          IN  NS   dns.2.com.
dns.2.com.      IN  A    192.168.1.2
```

for i in 'seq 1 253'; do echo "$i.com. IN NS dns.$i.com. dns.$i.com." >> com.db; echo "dns.$i.com. IN A 192.168.1.$i" >>com.db; done

子域

1. 88.com子域配置文件

vim /var/named/chroot/etc/named.conf

```
zone "88.com" IN { type master;
                   file "88.com.db"; };
```

2. 数据文件

vim /var/named/chroot/var/named/88.com.db

```
$TTL 86400
88.com.          IN  SOA  dns.88.com.  root\@88.com. (  )
88.com.          IN  NS   dns.88.com.
dns.88.com.      IN  A    192.168.1.88

88.com.          IN  MX   10  mail.88.com.
mail.88.com.     IN  A    192.168.1.88
```

chmod 640 88.com.db

chown root:named 88.com.db

service named restart

二. HOST

vim /etc/sysconfig/network

HOSTNAME = mail.88.com

vim /etc/hosts

```
127.0.0.1          localhost.localdomain    localhost
192.168.1.88       mail.88.com
```

reboot

vim /etc/resolv.conf

```
nameserver         192.168.1.254
```

三. MAIL

1. vim /etc/mail/sendmail.mc

   DAEMON_OPTIONS(Port=smtp, Addr=0.0.0.0, Name=MTA')dnl

2. vim /etc/mail/local-host-names

   88.com

   mail.88.com

3. cd /etc/mail

   make

4. service sendmail restart

5. sendmail -d0 #查看sendmail工作在哪个主机名上

四. TEST

1. 发邮件

   echo "haha~" | mail -s "nana" root@88.com

   echo "haha~" | mail -s "nana" root@[192.168.1.254] IP直发

2. 收邮件

   mail↙

   数字 #查看第n封邮件

   d1-3 #删除1-3封

   h #显示邮件标题列表

   q #退出

3. tail -f /var/log/maillog #查看日志.

4. mailq #显示邮件队列

No.

Date

```
cd /var/spool/mqueue
rm -rf *                                    #清空队列
```

五、aliases群发

```
vim /etc/aliases
最后一行
alluser:        nana,aa,bb                  #alluser随在,nana,aa,bb是本机存在的用户

newaliases                                  #hash化,数据转换,使生效
service sendmail restart

echo "haha~" | mail -s "nana" alluser@88.com
```

六、virtusertable虚拟账户转发

```
vim /etc/mail/virtusertable
virtuser@88.com         up@uplooking.com    #virtuser随在, tab隔开

make                                        #转换成virtusertable.db
service sendmail restart

echo "haha~" | mail -s "nana"
```

七、.forward

```
vim .forward或for i in 'seq 1 87';do echo "root@$i.com" >> .forward;done
chmod 644 .forward                          #进入用户主目录编辑.forward

echo "haha~" | mail -s "nana" user@88.com
```

八. access访问控制

vim /etc/mail/access

Connect:localhost.localdomain RELAY

Connect:localhost RELAY

Connect:127.0.0.1 RELAY

connect:192.168.1.88 OK,允许给本机任何一个用户发邮件,但不转发

connect:192.168.1.188 RELAY,允许转发

From:root@4.com REJEKT,拒绝

To:alluser@uplooking.com DISCARD,丢弃

make #转换成access.db

service sendmail restart

协议指令测试:

telnet 192.168.1.254 25 #ssh 192.168.1.88, telnet 254演示

helo mail.88.com

mail from:<root@88.com>

rcpt to:<up@uplooking.com>

data ↙正文

0

1

2

.结束并发送 #88发送至254成功

mail from:<root@88.com> #88 telnet 254转发给233 deny

rcpt to:<root@233.com>

九. SMTP验证

vim /etc/mail/sendmail.mc

52 TRUST_AUTH_MECH('EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl

53 define('confAUTH_MECHANISMS','EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl

make

service sendmail restart

/etc/init.d/saslauthd start

/etc/sysconfig/saslauthd # saslauthd 做验证

十. webmail 松鼠 squirrel /'skwirəl/

1. 安装包

squirrelmail-1.4.8-4.0.1.el5

2. cd /etc/httpd/conf.d

cat squirrelmail.conf

alias /webmail /usr/share/squirrelmail

3. cd /usr/share/squirrelmail/config/

./conf.pl

修改语言 zh_CH 保存

4. service httpd restart # http, php环境

5. http://192.168.1.88/webmail/src/webmail.php

非root帐户登录

+-. dovecot

vim /etc/dovecot.conf

21 protocols = imap pop3

service dovecot restart

vim /etc/services #所有服务对应的端口

netstat -ntltp #查看tcp端口是否开启

NIS

Date 2009 8 24

1.端口

2.安装包

ypserv-2.19-5.el5 # Server端

ypbind * Client端

3.配置文件

/var/yp/Makefile # Server端

/etc/nsswitch.conf * Client端

/etc/yp.conf

4.配置

(一)Server端

① hostname

nana #检查主机名,若重新设定主机名需修改

/etc/sysconfig/network和/etc/hosts

② echo "NISDOMAIN=88.com" >> /etc/sysconfig/network

echo "/bin/nisdomainname 88.com" >> /etc/rc.d/rc.local

#设置域名 开机启动

③ vim /var/yp/Makefile #定义要共享的信息.

all: passwd hosts

④ service portmap start

service ypserv start #启动服务

⑤ /usr/lib/yp/ypinit -m #生成数据库文件 ctrl+D保存

⑥ ls /var/yp/88.com #查看共享信息.

(二). Client端

① vim /etc/nsswitch.conf

hosts: nis files dns

② authconfig-tui

* NIS

domainname 88.com

NIS Servey 192.168.1.88

#相当于

tail -1 /etc/yp.conf

servise ypbind start

(三)测试

① Server端添加新用户aa, bb, 创建密码123

更新数据库 /usr/lib/yp/ypinit -m

② Client端

ypcat passwd

#能查看到Server端UID>500的用户passwd信息.

su - aa

#切换到普通用户aa

4. 解决普通用户可自己修改登录密码的问题

① Servey端

servise yppasswdd start

② Client端

su - aa

yppasswd

#新密码需大于6位

old:

new:

retry:

No

Date

5. 解决客户端主目录的问题

(一) Server端

① vim /etc/exports

/home 192.168.1.0/255.255.255.0(rw,sync)

#创建共享目录，读写，同步

service nfs start

② service autofs stop #关闭Server端自动挂载功能

chkconfig autofs off

③ vim /etc/auto.master

/home yp:auto.home

vim /etc/auto.home

* -rw,fstype=nfs 192.168.1.88:/home/&

④ vim /var/yp/Makefile

all: passwd hosts auto.master auto.home

⑤ /usr/lib/yp/ypinit -m

(二) Client端

① vim /etc/nsswitch.conf

automount: nis files

② service autofs start

③ su - aa

pwd

iptables

1. netfilter / iptables → 用户空间的管理命令
   ↓
   内核空间的功能机制

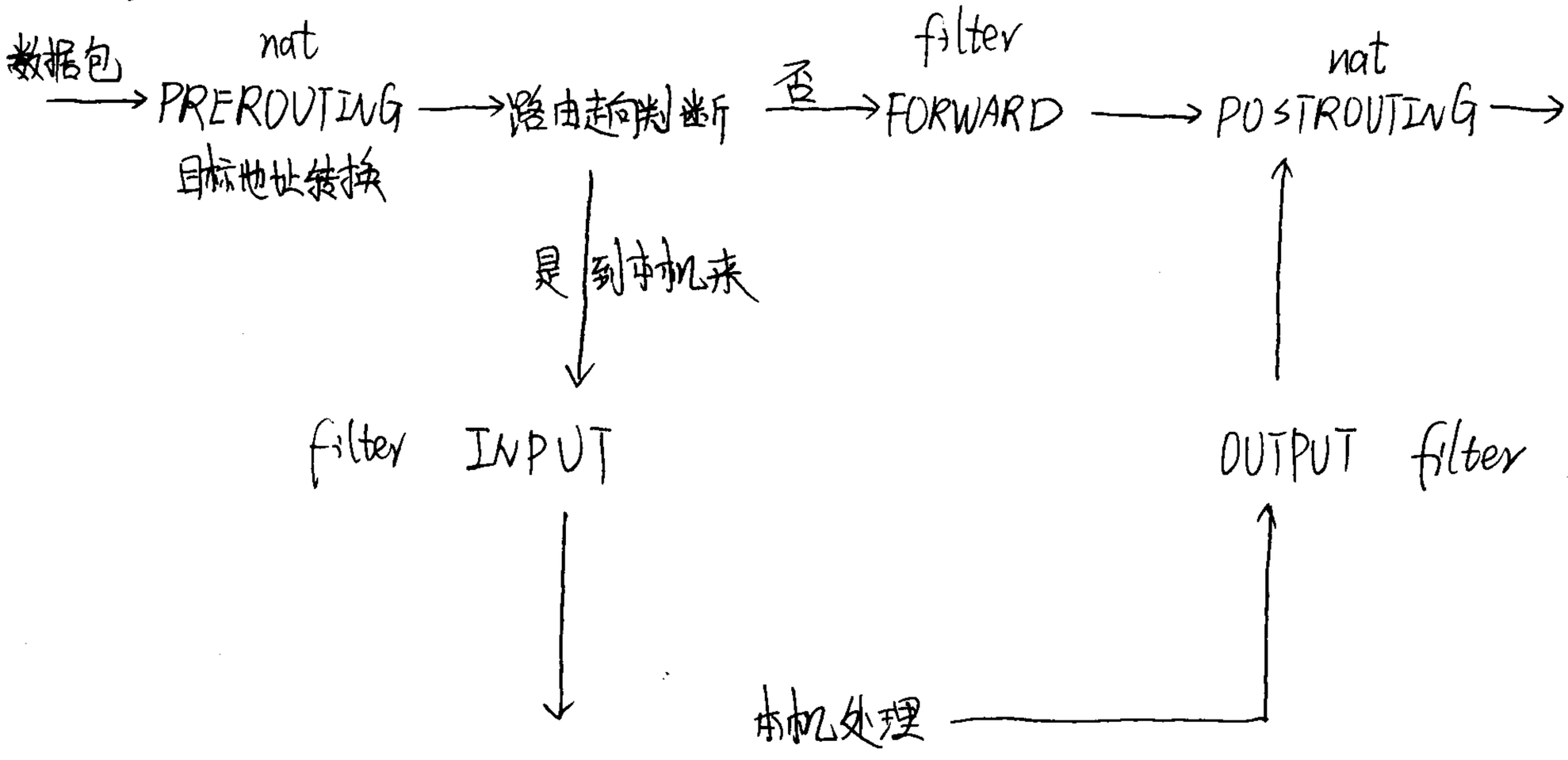
2. 表名　　链名

filter过滤表 ⊂ INPUT / OUTPUT / FORWARD

nat网络地址转换表 ⊂ PREROUTING / POSTROUTING / OUTPUT

mangle

3. 图示



4. 相关命令

(一) service iptables status　　#状态
service iptables start　　#启动
service iptables stop　　#关闭

(二) iptables -t filter -L　　#查看防火墙策略
iptables -t filter -L INPUT -v　　#查看INPUT策略
OUTPUT　　OUTPUT
FORWARD　　FORWARD
iptables -t filter -L INPUT --line-number
#查看策略编号
iptables -D INPUT 3　　#删除第三条策略
iptables -F　　#清空

service iptables save　　#保存,重启依然生效

(三) iptables -t 表名 -L 链名 编号 -p 协议 -s 源网段IP --sport 端口号
filter A INPUT 3 tcp 0.0.0.0/0
nat I OUTPUT udp
mangle D FORWARD icmp
F all
P

-d 目标网段IP　--dport 端口号　-i 网卡　-j ACCEPT
eth0 DROP
lo REJECT

#人:查看; A追加; I插入,插入编号3之前则成为编号3策略,不加编号插到第一条

```
iptables -t filter -A INPUT -p icmp --icmp-type echo-request
-s 192.168.1.88 -j ACCEPT
```

icmp请求ping，响应pong.

课后作业

1. 允许1个指定主机访问本地80和22端口，其他不行

```
iptables -A INPUT -p tcp -s 0.0.0.0/0 -d 192.168.1.88 --dport 80
-j DROP

iptables -A INPUT -p tcp -s 0.0.0.0/0 -d 192.168.1.88 --dport 22
-j DROP

iptables -I INPUT -p tcp -s 192.168.1.245 -d 192.168.1.88 --dport 80
-j ACCEPT

iptables -I INPUT -p tcp -s 192.168.1.245 -d 192.168.1.88 --dport 22
-j ACCEPT
```

2. 把默认策略设为drop，允许自己ping自己，所有人包括自己能访问80端口

```
iptables -P INPUT DROP

iptables -A INPUT -i lo -j ACCEPT

iptables -A INPUT -p tcp -s 0.0.0.0/0 -d 192.168.1.88
--dport 80 -j ACCEPT
```

3. 当默认策略drop以后，能访问别人的80和22端口

iptables -I INPUT -p tcp -s 192.168.1.0/24 --sport 80 -d 192.168.1.88 -i eth0 -j ACCEPT

iptables -A INPUT -p tcp -s 192.168.1.0/24 --sport 22 -d 192.168.1.88 -i eth0 -j ACCEPT

Shell 2009 8 25

一. bash 特殊字符

1. ` ` 反引号 #引用命令的执行结果，不能嵌套使用

a=`hostname` `echo ls` 相当于 ls

echo $a

server.uplooking.com

$( ) #与反引号作用相同，可嵌套使用

b=$(hostname) $(echo $(echo ls)) 相当于 ls

echo $b

server.uplooking.com

2. ~ #用户主目录

3. ! #取非

ls /dev/sda[!12]

/dev/sda3

#调用历史命令

!973

!ssh 最后一次以ssh开头的命令

!$ ls 对返回值取反(echo $? 查看返回值，0真，1-255假)

4. @

5. # #注释

6. $ #取变量的值

echo $PATH

/usr/lib/qt-3.3/bin:/

$( ) #引用命令的结果

No.

Date

$ { }　　#限定变量名的结束

　echo 2${a}0　　a=123

　21230

$ [ ]　　#运算

　echo $[1+2]

　3

7. %　　#取余,

　echo $[3%1]　0

　echo $[5%3]　2

　#

　sleep 30 &

　jobs　　[1]+

　kill %1 || kill %% || kill %+

8 ^　　#替换上条命令中的

　service vsftpd restart

　^vsftpd^httpd

9. &　　#后台任务

&&　　#逻辑与

　comm 1 && comm 2　前一个命令执行成功才执行后一个.

　make && make install

10. *　　#对文件名的扩展,任意长度任意字符

　ls /dev/sd*

　#乘法运算符

　echo $[5*2]

　10

11. ( ) 小括号 # 打开一个子shell

```
bb=123                (cc=111)
echo $bb              echo $cc
123                   无
                      (cc=111; echo $cc)
                      111
```

12 _ 下划线

13 - 减号 #减号运算符
#上一次目录

14. = #判值而非等于

15. + #加法运算符

16 | #管道

17. || #逻辑或
前一条命令失败才执行后一条，前一条命令成功则不执行后一条

18 { } 大括号 #枚举
touch {1,2,3,4}{a,b,c,d}
#命令列表
{ls -l ; echo 123 ; }

19. [ ] 中括号 #指定一个字符范围
ls /dev/sda[123
/dev/sda1 /dev/sda2 /dev/sda3
#作运算
$[ ]

等价于 test命令，作文件类型判断和数字比较

```
test 1 -eq 1          [ 2 -eq 1 ]
echo $?               echo $?
0                     1
```

20. \ 反斜杠 # 转义

```
mkdir a\ b        带有空格的目录名
# ls \↵ 换行
> /
```

21. : 冒号 # 表示某一个条件永远为真，内建命令，shell自动

外部命令，装包后产生的命令

```
: ↵
echo $? 查看返回值
0
```

22 ; 分号 #命令分隔符

```
comm1 ; comm2
```

23. " " 双引号 # 转义（弱）

```
mkdir "a b"   建立有空格的目录
```

24 ' ' 单引号 # 转义（强）

```
echo '!!'
```

25. <

>

>>

<< #打开临时缓冲区 cat：把输入原封不动输出

```
cat > file << EOF
>
>EOF 结束并予输出
```

26. ，逗号　　#枚举时的分隔
　　touch {1,2,3,4} {a,b,c,d}

27 . 点　　#内建命令，当前shell，等价于source

28 ?　　#单个任意字符

29. / 斜杠　　#根
　　#目录分隔符
　　////=/

30. ↵ 回车　　#执行命令

31. ␣ 空格　　#命令与选项之间的分隔

思考：

mkdir {a,b,c,d}{1,2,3,4}
ls
在后面加 .sh
find ./* -exec mv {} {}.sh \;　　#{}找到的文件名，{}引用
　　\转义
　　;分隔命令

去掉.sh
for i in ./*; do mv $i `echo $i | cut -d. -f 1,2`; done

二. sed

(一) 流编辑器,对文件输出的过程进行操作,逐行输入和输出

sed -选项 '定址命令' 文件名

-n	行号	d
-i	/正则/	p
-f	s///g	
	y///	
	i	
	a	
	w	
	r	
	=	
	h	
	H	
	g	
	G	
	x	

#单引号,不写定址表示所有行
#删除
#打印,配合-n使用关闭原始模板输出
#替换 /old/new/
#变形,前后一一对应
#在定址前加注解
#在定址后加注解
#另存为
#从文件中读
#显示行号

#-n,关闭原始模板输出
#-i,修改原文件
#-f,从文件中调用命令

删除实例

1. sed '1d' /etc/passwd
   '1,30d'

#删除第1行
#删除第1~30行 , 1,$, 1~最后一行

2. sed '/bash$/d' /etc/passwd
   '/^root/d'

#删除以bash结尾的行
#删除以root开头的行

3. sed '/^root/,/^user/d' /etc/passwd

#删除第1个以root开头和第1个user开头之间的行

sed '/^root/,54d' /etc/passwd

#正则与行号混合使用

打印实例

1. sed '55p' /etc/passwd　　#打印第55行，显示两行

   sed -n '55p' /etc/passwd　　#显示一行

   sed -n '/^root/p' /etc/passwd　　#打印以root开头的行

   grep '^root' /etc/passwd

替换实例

1. sed 's/[a-z]/(&)/g' /etc/passwd　　#将所有小写字符加小括号 &引用

2. vim a

   xxx root xx pg xxx

   .* root .* pg .*

   1 2 3 4 5

使root与pg替换

   sed 's/\(.*\)\(root\)\(.*\)\(pg\)\(.*\)/\1\4\3\2\5/g' /etc/passwd

   #\(\)标记，引用

   sed -r 's/(.*)(root)(.*)(pg)(.*)/\1\4\3\2\5/g' /etc/passwd

No

Date

(二) sed高级定址、命令

变形实例

1. sed 'y/abcd/ABCD/' #把abcd变形为ABCD,一对应

2. sed 'y/abcdefghijklmnopqrstuvwxyz/fghijklmnopqrstuvwxyzabcde/'
#简单加密

3. vim bx.sed
#!/bin/sed -f
y/ / /

chmod +x bx.sed
./bx.sed /etc/passwd

vim bx.cat #自显示脚本
#!/bin/cat

vim bx.rm #自删除脚本
#!/bin/rm

注释实例

1. sed '/^root:/i He is an administrator' /etc/passwd
#给root用户添加注释,显示在定址的上行

2. sed '/^root:/a␣He is an administrator' /etc/passwd
#注释显示在定址后,下一行

这里无␣

另存为实例

1. sed '1,5w /root/test' /etc/passwd　　#将passwd第1~5行保存至test文件覆盖.

读取文件实例

1. sed '5r a' /etc/passwd　　#将a文件中的内容读到passwd第55行后

打印偶数行实例　p, d

1. sed -n '0~2p' /etc/passwd　　#从0开始,步长为2,~表示步

sed '=' /etc/passwd　　#显示行号

sed -n '0~2{p;=}' /etc/passwd　　#相同定址不同命令,命令用;隔开 {p;=}命令列表

相同定址不同命令实例

1. sed -n '0~2{p;=}' /etc/passwd

不同定址相同命令实例

1. sed '/^#/d;/^$/d' /etc/inittable　　#/^#/注释, /^$/空行多点编辑

2. sed -e '/^#/d' -e '/^$/d' /etc/inittab

#删除注释，删除空行，同1

3. sed -n '/^root/p ; /^bin/p' /etc/passwd

#打印以root和bin开头的行

4. sed '/nologin$/d ; /bash$/d' /etc/passwd #删除以nologin和bash结尾的行

(三)模式空间和暂存空间

作用在暂存空间的命令 h, H, g, G, x ，暂存空间默认有一个单独的空行

实例

1. sed '1h;8G' a #把第1行暂存，追加到模式空间第8行后面

2. sed '1h;2H;3x;4x;8G' a

vim a

111
222
333
444
555
666
777
888

执行命令后的结果

111
222
111
222
333
555
666
777
888
444

思考：

一. mkdir {a,b,c,d}{1,2,3,4}

ls

a1 a2 a3 a4 b1 b2 b3 b4 c1 c2 c3 c4 d1 d2 d3 d4

二. for i in ./* ;

do mv $i `echo $i | sed 's/.*/&.sh/g' ` ;

done

ls

a1.sh a2.sh a3.sh a4.sh b1.sh b2.sh b3.sh b4.sh
c1.sh c2.sh c3.sh c4.sh d1.sh d2.sh d3.sh d4.sh

三. for i in ./* ;

do mv $i `echo $i | sed 's/\.sh//g' ` ;

done

ls

a1 a2 a3 a4 b1 b2 b3 b4 c1 c2 c3 c4 d1 d2 d3 d4

或者 sed 's/\(.*\)\(\.sh\)/\1/g'

sed 's/\.[a-z]*$//g'

sed -r 's/(.*)(\.[a-z]*)/\1/g'

No.

Date

三. 正则 regular express

vim, grep, sed, pel, awk

(一) grep 基本正则

^root	#以root开头的行
bash$	#以bash结尾的行
\<ro	#以ro开头的单词
sh\>	#以sh结尾的单词
.	#任意单个字符
[a-zA-Z0-9]	#指定范围的单个字符
*	#重复前1个字符任意次(包括0)
.*	#任意长度任意字符
[^]	#取非
^[^#]	#不以#开头的行
[^:]*	#不是:的任意长度字符
a\{n,m\}	#表示重复前1个字符a, n~m次
a\{1,3\}	#3次以下 或 1到3次
a\{3,\}	#3次以上
a\{3\}	#3次

实例

grep '^[^:]*:[^:]*:[0-9][0-9[0-9]:' /etc/passwd
#UID≥100 三位数

grep '^[^:]*:[^:]*:[0-9]\{3,\}:' /etc/passwd

(二) egrep 扩展正则 或 grep -E

?	#重复前1个字符0次或1次
+	#重复前1个字符1次或1次以上
{n,m}	#重复前1个字符n到m次
(ro)+	#重复ro1次或1次以上
(ro){2}	#重复ro2次
	(ro) 字符模式

实例

1. egrep '^root|^bin' /etc/passwd　　#以root开头或'bin开头

   ||

   '^(root|bin)'

2. egrep '^root|bin$' /etc/passwd　　#root开头或bin结尾

3. sed -r

(三) fgrep 快速正则

fgrep ^a　　#过滤a文件中^的行,不解释特殊含义
过滤特殊字符时用到

(四) pgrep

pgrep httpd　　#显示httpd进程pID.

No.

Date

```
[a-zA-Z0-9]\{1,\}                                  #单词

[^a-zA-Z0-9]\{1,\}                                 #单词分隔符

sed 's/^.//g' /etc/passwd                          #删除每行第1个字符

sed 's/\(^.\).//\1/g'                              #删除每行第2个字符

sed 's/.$//g'                                      #删除每行最后一个字符

sed 's/.\(.$\)/\1/g'                               #删除每行倒数第2个字符

date +%m/%y/%d | sed 's#/#;#g'                     #把 mm/yy/dd的日期格式转换为
                                                     mm;yy;dd

sed -r 's/[a-z]*[A-Z]+[a-z]*/(&)/g' /etc/passwd    #把所有含大写字母的单词加小括号
```

awk

Date 2009 8 26

awk，报表生成器，逐行输入和输出，对列操作

条件
/正则/

awk -F: 'BEGIN{OFS="-"; ORS="!"; printf("%-15s%-7d\n", $1, $3)} 定址

-F指定分隔符　默认为空格

OFS：输出字段分隔符

ORS：输出记录分隔符

%s：字符串　　, 输出分隔符为空格

%d：数字　　"　" 多个空格

-15：左对齐　　"\t" TAB

默认右对齐

\n：换行

print默认不换行，printf默认换行

{命令} END{print "　　"}'　　filename

BEGIN：行前操作

END　：行尾操作，记忆最后一行

$0　　#一整行

$1, $2, $3　　#第1列，第2列，第3列

NR　　#行号，记录号

NF　　#字段数

一、print

1. awk -F: '{print $0}' /etc/passwd　#不写定址默认所有行，相当于{print}

2. awk -F: '{print $1, $3}' /etc/passwd #打印用户名和UID

3. df -h | awk '{print $1"\t"$5}' #打印第1列和第5列，输出字段用TAB分隔

4. awk -F: '{print NR}' /etc/passwd　　#显示每行行号 only

awk -F: '{print NR, $0}' /etc/passwd

5. awk -F: '{print NF}' /etc/passwd　　#显示每行字段数

awk -F: '{print $0, NF}' /etc/passwd

6. awk -F: '/^root/{print NR, $0, NF}' /etc/passwd

#输出行号，行内容和这行字段数.

正则定址

awk '/^root/{print}' /etc/passwd　　#输出以root开头的行

条件定址

(一)比较运算符

==	#等于
!=	#不等于
<	#小于
<=	#小于等于
>	#大于
>=	#大于等于
~	#匹配
!~	#不匹配

```
awk 'NR==5{print}' /etc/passwd          #输出第5行

awk -F: '$3==0{print}' /etc/passwd      #输出UID=0的行

awk -F: '$3>500{print}' /etc/passwd     #输出UID>500的行

awk -F: '$3==0{print $1,$3}' /etc/passwd  #输出UID=0的用户名和UID.
```

```
awk -F: '$1=="root"{print}' /etc/passwd

awk -F: '$1~/^root/{print}' /etc/passwd  #正则匹配

awk -F: '$3~/^5/{print}' /etc/passwd
```

(二)逻辑运算符

```
&&          #与
||          #或
!           #非
```

```
awk -F: '$3>=500 && $3<=505{print}' /etc/passwd
                #输出 500≤UID≤505的行
```

二. printf 格式化输出

```
awk -F: '{printf("%-15s%-7d\n",$1,$3)}' /etc/passwd
```

```
#printf(  ,  )
  %s 描述字符串
  %d 描述数字
  %15s ,15是偏移量,默认右对齐
```

※ 每隔5行打印标题

```
vim  title.awk
#!/bin/awk -f
BEGIN{
        FS=":"
        printf ("%-15s%-7s\n","Username","UID")
}
{
        if(count<5){
                printf("%-15s%-7d\n",$1,$3)
                count++
                } else {
                count=1
                printf("%-15s%-7s\n","Username","ULD")
                printf("%-15s%-7d\n",$1,$3)
                }
}
END{
        print "done"
        system("date")
}
```

No

Date

if条件语句

1. 输出奇数行 vim 1.awk

```
#!/bin/awk -f
{
        if(NR%2==1){
                print NR,$0
        }
}
```

else if
else

2. 输出偶数行 vim 2.awk

```
#!/bin/awk -f
NR%2==0{
        print NR,$0
}
```

while循环语句

1、输出0-10 vim while.awk

```
#!/bin/awk -f
BEGIN{
count=0
while(count<=10){
        print count
        count++
}
}
```

for循环语句

1. 输出0－10 vim for.awk

```
#!/bin/awk -f
BEGIN{
        for(count=0;count<=10;count++){
                print count
        }
}
```

2. 找出 11000 ~ 12000的质数 vim zs.awk

```
#!/bin/awk -f
BEGIN{
        for(i=11000;i<12000;i++){
                for(j=2;j<i/2;j++){
                        if(i%j==0){
                                ret=0
                                break
                        }else{
                                ret=1
                                continue
                        }
                }
                if(ret==1){
                        print i
                }
        }
}
```

stu1	82	56
stu2	70	25
stu3	87	31
stu4	68	69
stu5	70	59

1. 求每个学生语文数学的平均成绩 score.awk

```
#!/bin/awk -f
{
        print $1,$2,$3,($2+$3)/2
}
```

2. 求全班学生的语文平均成绩和数学平均成绩 score2.awk

```
#!/bin/awk -f
BEGIN{
        count=0
        chin=0
        math=0
}
{
        count++
        chin=$2+chin
        math=$3+math
}
END{
        print "chin_avg:" chin/count
        print "math_avg:" math/count
}
```

添加用户 system.awk

```
#!/bin/awk -f
{
        system("useradd "$1";echo "$2" |passwd --stdin "$1)
}
```

统计系统中Uid≥500的用户数 usercount.awk

```
#!/bin/awk -f
BEGIN {
        FS=":"
        count=0
}
$3>=500{
        count++
}
END{
        print "user count: " count
}
```

```
vim sshlog.awk
#! /bin/awk -f
{
        for(i=1;i<=NF;i++){
                if($i~/^sshd/&&$NF~/user=root$/){
                        print $(NF-1),$NF
                }
        }
}
```

No.

Date

1. time ./zs.awk #执行脚本并显示执行时间 real

2. passwd --stdin user1 #提示输入密码

echo "123" | passwd --stdin user1

3. system("date") #system("命令")

system("useradd "$1)

awk -F: '$3>=500 && $3<=60000{system("userdel -r "$1)}' /etc/passwd

#删除系统中>500的用户

倒序排列文件的所有字段

```
#!/bin/awk -f
BEGIN{
        FS=":"
}
{
        for(i=NF;i>0;i--){
                if(i!=1){
                        printf("%s:",$i)
                }else{
                        printf("%s",$i)
                }
                print ""        #换行
        }
```