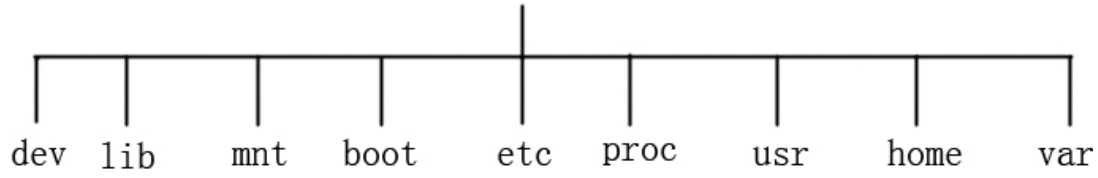
第一章文件系统及常用命令



1.1

1.1linux 目录结构和文件系统

/bin 二进制可执行命令
/dev 设备特殊文件
/etc 系统管理和配置文件
/home 用户的宿主目录，也称家目录
/lib 标准程序设计库，又叫动态链接共享库，作用类似 windows 里的.dll 文件
/sbin 系统管理命令，这里存放的是系统管理员使用的管理程序
/tmp 公用的临时文件存储点(系统重启后会清空)
/root 系统管理员的主目录（领导的目录）
/mnt 系统提供这个目录是让用户临时挂载其他的文件系统。
/proc 虚拟的目录，是系统内存的映射。可直接访问这个目录来获取系统信息。
/var 某些大文件的溢出区，比方说各种服务的日志文件
/usr 最庞大的目录，要用到的应用程序和文件几乎都在这个目录。其中包含：
/usr/bin 众多的应用程序
/usr/sbin 超级用户的一些管理程序
/usr/doc linux 文档
/usr/include linux 下开发和编译应用程序所需要的头文件
/usr/lib 常用的动态链接库和软件包的配置文件
/usr/man 帮助文档

补充说明：

1．/bin 目录包括了系统常用的 Linux 命令。任何用户都可以直接运行这些命令而无需指定环境。

2. ./dev 目录包括了特殊设备的特殊文件，由于 Linux 中把所有的外部设备都看作是文件处理，该目录里面文件被看作一个外部设备。

3. /etc 目录中包含各种各样的系统程序和数据文件。例如在/etc/rcx.d 中包含了启动不同级别要执行的脚本，初始化的环境。

4．/lib 目录包括编程语言运行所需库文件。这个目录很重要是必须的。

5．/usr/include 编译 C 语言的头文件。

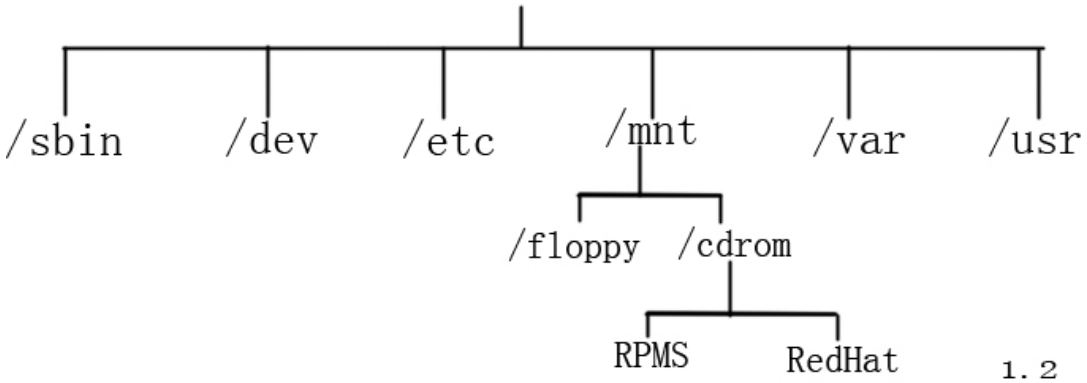
6．/tmp 目录中保存了 Linux 系统程序产生的临时文件。这些文件一般只在相应的程序正在运行时才存在，如果程序非正常终止，这些临时文件，可能会保存在/tmp 目录中，我们可以删除任何不属于正在运行的临时文件。

1．1．1 linux 文件系统

文件系统在组织成树行结构后，文件名则被指定由文件系统的根目录开始到指定的文件位置的完整路径，以完整路径表示的为绝对路径，此外表示路径的方法还有“相对”路径，相对路径使用两种路径标识：“.”表示本身，即当前目录，“..”表示上一级目录。另外 linux\unix 有很多不同于其他操作的特征：

1. 把所有的外部设备看作是文件，为了保证输入输出操作的一致性 linux\unix 把所有的对外部设备的操作都设计成为等同于操作文件。因此无论什么外部设备的特征，都被文件系统隐藏了。用户可以使用与文件系统相同的系统调用和函数来读写硬件设备。这些代表硬件设备的文件存在/dev 目录下边。他们只占用系统的逻辑位置，而不占用实际的物理磁盘存储空间，对/dev 目录里的文件读写，都被透明的传递给相应的外部设备，从而方便，透明，高效的进行外部设备控制。
2. 文件的用有者可以指定其他的组或用户读，写和执行文件的权限，从而起到提高文件的安全性。
3. 文件系统可以合并。Linux 文件系统是一个逻辑的概念，从面上看是一个完整的整体，实际上是由一个或者是多个独立的目录树组成；每一个目录树都是一个独立的文件系统，这种结构，完全跨越了物理存储设备的限制，可以把多个存储设备和多个不同的文件系统共同组成一个文件系统。要使用其他的文件系统，必须把他们安装到自己的文件系统目录去。安装的位置被称为“装载点”。

   文件系统安装后，和原来的文件系统构成了一个新的文件系统。被装载的文件系统，覆盖了装载点原来的目录。当被安装的文件系统卸载后，原来的目录就被恢复，例如图 1.2 将光盘安装到/mnt/cdrom 目录中。



1.2

1．2 文件的类型

Linux 系统中有三种基本的文件类型：普通文件、目录文件和设备文件。

a. 普通文件:是用户最经常面对的文件。它又分为文本文件和二进制文件。

1)文本文件：这类文件以文本的 ASCII 码形式存储在计算机中。它是以"行"为基本结构的一种信息组织和存储方式。

2)二进制文件：这类文件以文本的二进制形式存储在计算机中，用户一般不能直接读懂它们，只有通过相应的软件才能将其显示出来。二进制文件一般是可执行程序、图形、图像、声音等等。

b. 目录文件:主要目的是用于管理和组织系统中的大量文件。它存储一组相关文件的位置、大小等与文件有关的信息。目录文件往往简称为目录。

c. 设备文件:是 Linux 系统很重要的一个特色。Linux 系统把每一个 I/O 设备都看成一个文件，与普通文件一样处理，这样可以使文件与设备的操作尽可能统一。从用户的角度来看，对 I/O 设备的使用和一般文件的使用一样，不必了解 I/O 设备的细节。设备文件可以细分为块设备文件和字符设备文件。前者的存取是以一个个字符块为单位的，后者则是以单个字符为单位的。

补充说明：
字符设备：只的是那写一次性输入输出操作，仅能处理一个自己的数据设备。例如终端
块设备：指的是通过大小固定的缓冲区进行输入输出操作的，也可以在一次操作中处理多个字符设备。

1．3 文件属性

在 linux/Unix 中，每一个文件都有一个 16 位的字段来记录他们的属性，用户 ID，组 ID 存储权限等信息。16 位字段作用如下：

属性位	符号含义
1-8	所有者 r-w-x, 组用户 r-w-x, 其他用户 r-w-x
9	粘着标志位。
10	用户组标志。
11	用户标志符号。
12	命名管道。
13	字符和块设备。
14	目录
15	一般文件

文件 1-8 位所代表的含义，如下图 2-3 所示

R	W	X	R	W	X	R	W	x
读	写	执行	读	写	执行	读	写	执行
所有者权限			组用户的权限			其他用户的权限		

2-3

r-w-x 含义

	r	w	x
一般文件	可以读文件的内容	可修改文件	可执行文件
目录	可显示目录内容	可将数据写入目录	可以包括搜索路径
设备文件	可从设备读数据	可将数据写入设备	无意义

2-4

属性中的第 9 位为粘着位（sticky）,粘着为 1 时，文件被装入内存后将一直保留在内中直到系统的关闭。管理员可以对一些使用率高，运行速要求高的程序设置粘贴位，以减少系统内的等待时间。但设置多了会占用内存增加系统的负担，反而影响系统速度。请适当设置 。属性位中的第 10 位是用户标识，即一个用户一个 ID，一个组 ID，系统中默认有 root，backup 等组，这些组被赋予了不同的权限，创建组用户时候继承了组的权限。我们可以以为新创建的用户指定组，也可以为已经存在的用户修改其所在的组，属性位的第 11 位为粘着标志，下面举例说明 SUID 的概念。;

在一个网络中，构件了 linux/uninx 操作系统服务，员工的报告通过网络传输到领导的账号中，员工可以拷贝命令将报告付制到领导的目录中。但是这样会有一些问题，如果领导指定了某个为接受数据的目录，那么这个目录必须员工拥有写权限，但是没有可读权限，这是为了防止，员工与员工之间互相察看数据，而员工不能看见目录中的信息，很可能在写的过程覆盖。

解决这个问题就用到了 SUID 位，员工文件如果可以被设置了启动 SUID 位，那么执行该文件的用户在执行过程中，就具有和文件所有者一样的权利。如：在领导账号中创建另一个员工的数据文件，数据文件对领导可读可写，而对其他用户没有任何读，写执行权限，默认情况下，其他员工不能读取，修改或删除数据文件。同时领导创建另一个写入文件 sumit，该文件用于向上面的数据文件添加，修改数据，同时设置 sumit 的 SUID 位，这样员工虽然不能直接读写领导目录中的数据，但是可以通过执行 sumit，获得权限将数据写入领导的目录中。

在第 12-15 位中指定了文件类型，是在文件创建的时候。用户不能修改。由于文件职能是这 4 种类型之一，所以这 4 位中只可能有一位是 1。

Linux 通过 r-w-x 来控制用户以及组权限，通常修改文件的权限有两种方法一种是符号模式，另一种是绝对模式。

1 符号模式的语法：Chmod [who] operator [permission] filename
Who (u,g,o,a)　u：本身用户 g：组用户 o：其他用户 a：代表所有用户
Operator (+,-,=)　+：表示增加　-：表示减去　=：付值
Permission (r,w,x,s,t)　r：写权限　w：写权限　x：执行权限　s：SUID t:拥有这才能删除
例：　chmod　o+w　/etc/php.ini

绝对模式 chmod [1,2,4]filename
R=4　w=2　x=1
例: chmod 755 /etc/php.ini

1.4Linux 系统常用命令及语法

1) 显示文件及文件目录 ls--(list)

```
[wds@localhost ~]#  ls    参数          #常用参数和意义
[wds@localhost ~]#  ls     -a           #显示目录下所有文件包括隐藏文件
[wds@localhost ~]#  ls     -l           #以列表显示
例:
[wds@localhost ~]#   ls -al
total 188
drwxr-x--- 14 root root  4096 Oct 25 21:53 .
drwxr-xr-x 25 root root  4096 Nov 10 21:04 ..
-rw-------  1 root root  1069 Oct 25 21:41 anaconda-ks.cfg
-rw-------  1 root root    31 Oct 25 21:46 .bash_history
-rw-r--r--  1 root root    24 Jan  6  2007 .bash_logout
```

2) 切换目录命令 cd

```
[wds@localhost ~]# cd     参数            # 常用参数和意义
[wds@localhost ~]# cd       -             #退回到刚刚去的目录
[wds@localhost ~]# cd       ~             #返回宿主目录
[wds@localhost ~]# cd       ..            #去上一级目录
例:
[wds@localhost ~]# cd     /etc            #进入/etc 目录
[wds@localhost ~]# cd     ~               #返回宿主目录
```

3) 创建目录命令 mkdir(make directories)

```
[wds@localhost ~]# mkdir  参数                  # 常用参数和意义
[wds@localhost ~]# mkdir  test                  #建立目录
[wds@localhost ~]#  mkdir  -p /test/data/a      #循环建立目录
例:
[wds@localhost ~]# mkdir test
[wds@localhost ~]#  ls
anaconda-ks.cfg  Desktop  install.log  install.log.syslog  test
[wds@localhost ~]# mkdir -p .a/b/c/d
[wds@localhost ~]# tree
.
`-- a
    `-- b
        `-- c
            `-- d
```

4) 显示当前目录命令 pwd(print working directory)

```
[wds@localhost ~]# pwd    参数                  # 常用参数和意义
[wds@localhost ~]# pwd                          #显示当前位置
例:
[wds@localhost ~]# pwd
/root
```

5) mv 移动与修改

```
[wds@localhost ~]# mv   参数                    # 常用参数和意义
[wds@localhost ~]# mv    -i                     # 覆盖前先询问用户

[wds@localhost ~]# mv  oldname  newname         #改文件名
例:
[wds@localhost ~]# mv test  test1
[wds@localhost ~]#  ls
anaconda-ks.cfg  Desktop  install.log  install.log.syslog  test1

[wds@localhost ~]# mv  filename  target         # 移动目录
例:
[wds@localhost ~]# mv  test1   /etc
[wds@localhost ~]#  ls
anaconda-ks.cfg  Desktop  install.log  install.log.syslog
```

6) find 查找命令

```
[wds@localhost ~]# find   参数                    # 常用参数和意义
[wds@localhost ~]# find  -name        文件名      # 查找文件名
[wds@localhost ~]# find  -type       (f|d|c|l)    # 查找文件类型
[wds@localhost ~]# find  -exec                    # 执行系统命令
[wds@localhost ~]# find  -mindepth                # 目录级
[wds@localhost ~]# find  -mtime                   # 目录级


例:
[wds@localhost ~]# find .  -name "*.conf"     # 查找当前目录下，以 conf 结尾文件
./pam_smb.conf
./java/java.conf
./nsswitch.conf
./gconf/2/evoldap.conf
./alsa/alsa.conf
./alsa/pcm/side.conf
./alsa/pcm/front.conf
[wds@localhost ~]# find /var/log -type f -mtime +20 |xargs rm -f # 查找 20 天前日志并删除
```

7) cp 复制命令

```
[wds@localhost ~]# cp  参数                    # 常用参数和意义
[wds@localhost ~]# cp  目标源文件   目标文件     # 查找当前目录下，以 conf 结尾文件
[wds@localhost ~]# cp  -R                      # 复制目录

例:
[wds@localhost ~]# cp  -R test  test1
[wds@localhost ~]# ls
anaconda-ks.cfg  Desktop  install.log  install.log.syslog  test1  test
```

8.) 显示文件内容命令 cat

```
[wds@localhost ~]# cat   参数                  # 常用参数和意义
[wds@localhost ~]# cat    -A     file          # 显示特殊字符
[wds@localhost ~]# cat    -n     file          # 显示行号

例:
[wds@localhost ~]# cat -n /etc/passwd
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
```

9) wc 显示文件大小（行数，字数）

```
[wds@localhost ~]# wc   参数                # 常用参数和意义
[wds@localhost ~]# wc    -l       file      # 显示多少行
[wds@localhost ~]# wc    -w       file      # 显示多少单词
[wds@localhost ~]# wc    -c       file      # 显示字数

例:
[wds@localhost ~]# wc -l /etc/passswd
[wds@localhost ~]# 35
```

10) 文件内容排序 sort

```
[wds@localhost ~]# sort   参数                # 常用参数和意义
[wds@localhost ~]# sort    -n       file      # 数学形式排序
[wds@localhost ~]# sort    -r       file      # 倒序排列
[wds@localhost ~]# sort    -n       file      # 显示行号
[wds@localhost ~]# sort    -u       file      # 排重
```

11) 在文中查找命令 grep

```
[wds@localhost ~]# grep   参数                          # 常用参数和意义
[wds@localhost ~]# grep  "查找内容"          file       # 查找文件内容
[wds@localhost ~]# grep    -l   "查找内容"   file       #  显示要查找内容的文件名
[wds@localhost ~]# grep    -n   "查找内容"   file       # 显示行号
[wds@localhost ~]# grep    -数字  "查找内容"  file       # 显示查找内容的上下行内容

例:
[wds@localhost ~]# grep "root" /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
```

12)文件截取命令 cut

```
[wds@localhost ~]# cut   参数                    # 常用参数和意义
[wds@localhost ~]# cut    -d  "分隔符"   file      # 查找文件内容
[wds@localhost ~]# cut    -f  "列名"     file      # 常与-d 参数连用，用些分列显示列名
[wds@localhost ~]# cut    -c  "数字"     file      # 显示截取后的内容
例:
[wds@localhost ~]# cut -d : -f1 /etc/passwd
root
bin
daemon
```

13) 查看系统负载 top

```
[wds@localhost ~]#top
                Tasks: 29 total  进程总数
                1 running  正在运行的进程数
                28 sleeping  睡眠的进程数
                0 stopped  停止的进程数
                0 zombie  僵尸进程数
                Cpu(s): 0.3% us  用户空间占用 CPU 百分比
                1.0% sy  内核空间占用 CPU 百分比
                0.0% ni  用户进程空间内改变过优先级的进程占用 CPU 百分比
                98.7% id  空闲 CPU 百分比
                0.0% wa  等待输入输出的 CPU 时间百分比
                0.0% hi
                0.0% si

                Mem: 191272k total  物理内存总量
                173656k used  使用的物理内存总量
                17616k free  空闲内存总量
                22052k buffers  用作内核缓存的内存量
                Swap: 192772k total  交换区总量
                0k used  使用的交换区总量
                192772k free  空闲交换区总量
                123988k cached  缓冲的交换区总量。
                内存中的内容被换出到交换区，而后又被换入到内存，但使用过的交换区尚
                未被覆盖，
                该数值即为这些内容已存在于内存中的交换区的大小。
                相应的内存再次被换出时可不必再对交换区写入。

                进程信息区统计信息区域的下方显示了各个进程的详细信息。首先来认识一下
                各列的含义。
                序号  列名  含义
                a PID  进程 id
                b PPID  父进程 id
                c RUSER Real user name
                d UID  进程所有者的用户 id
                e USER  进程所有者的用户名
                f GROUP  进程所有者的组名
                g TTY  启动进程的终端名。不是从终端启动的进程则显示为 ?
                h PR  优先级
                I NI nice 值。负值表示高优先级，正值表示低优先级
                j P  最后使用的 CPU，仅在多 CPU 环境下有意义
                k %CPU  上次更新到现在的 CPU 时间占用百分比
                l TIME  进程使用的 CPU 时间总计，单位秒
                m TIME+  进程使用的 CPU 时间总计，单位 1/100 秒
```

```
n %MEM  进程使用的物理内存百分比
o VIRT  进程使用的虚拟内存总量，单位 kb。VIRT=SWAP+RES
p SWAP  进程使用的虚拟内存中，被换出的大小，单位 kb。
q RES  进程使用的、未被换出的物理内存大小，单位 kb。RES=CODE+DATA
r CODE  可执行代码占用的物理内存大小，单位 kb
s DATA  可执行代码以外的部分(数据段+栈)占用的物理内存大小，单位 kb
t SHR  共享内存大小，单位 kb
u nFLT  页面错误次数
v nDRT  最后一次写入到现在，被修改过的页面数。
w S  进程状态。
D=不可中断的睡眠状态
R=运行
S=睡眠
T=跟踪/停止
Z=僵尸进程
x COMMAND  命令名/命令行
y WCHAN  若该进程在睡眠，则显示睡眠中的系统函数名
z Flags  任务标志，参考 sched.h
```

[

14) 管道

```
[wds@localhost ~]#ps –ef | cut –c1-10 | sort > log
```

15) 查看内存 free

```
[wds@localhost ~]#free -m
             total       used       free     shared    buffers     cached
Mem:          4051       1872       2178          0        100       1564
-/+ buffers/cache:        207       3843
Swap:         4094          0       4094
```

16) 查看系统硬盘空间 df

```
[wds@localhost ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/cciss/c0d0p1     3.9G  669M  3.1G  18% /
/dev/cciss/c0d0p7     104G  198M   98G   1% /data0
tmpfs                 2.0G     0  2.0G   0% /dev/shm
/dev/cciss/c0d0p5     3.9G  317M  3.4G   9% /tmp
/dev/cciss/c0d0p2     9.7G  3.0G  6.2G  33% /usr
/dev/cciss/c0d0p3     7.8G  1.7G  5.7G  23% /var
```

1.5 vi 的使用

命令模式（进入时进入是的默认模式）：
任何输入都会作为编辑命令，而不会出现屏幕上，若输入错误则有声音提示
输入模式（编辑模式：
任何输入的数据都置与编辑寄存器。在命令模式下输入（i，a，A 等），可以进入
输入模式，输入 ESC 可以返回命令模式
特殊模式（最后行模式）：
以“:”或者“/”为前导指令，出现屏幕的最下一行，任何输入当作命令执行。

命令	输入的方法
进入输入的方法	
<a>	在光标的后边输入文本。
<A>	在当前行末尾输入文本。
<i>	在光标前输入文本。
<I>	在当前行开始输入文本。
<o>	在当前行后输入新一行。
<O>	在当前行前输入新一行。
光标移动的方法	
<b>	移动到当前单词开始。
<e>	移动到当前的单词尾开始。
<w>	向前移动一个单词。
<h>	向前移动一个字符。
<j>	向上移动一行。
<k>	向下移动一行。
<l>	向后移动一个字符。
删除操作	
<x>	删除光标所在的字符。
<dw>	删除光标所在的单词。
<d$>	删除光标至行尾所有字符。
<dd>	删除当前行。
改变与替换操作	
<r>	替换光标所在字符。
<R>	替换字符序列。
<cw>	替换一个单词。
<cb>	替换光标所在的前一个字符。
<c$>	替换自光标位置至尾行的所有字符。
查询命令	
</abc>	向前查询 abc
<?abc>	向后查询 abc
<n>	向前查询
<N>	向后查询

拷贝粘贴命令	
</yw>	将光标所在的单词拷贝到剪贴板
<y$>	将光标至尾的字符拷贝如剪贴板
<yy>	将当前行拷贝到剪贴板
<p>	将剪贴板的内容粘贴到光标后
<P>	将剪贴板的内容粘贴到光标前
文件保存及退出 vi	
<:q>	不保存退出。
<:q!>	不保存强制退出。
<:w>	保存编辑。
<:w filename>	存入文件 filename 中。
<:w! filename>	强制存入 filename 中.
<:wq>	保存退出
其他命令	
<:set nu>	显示行号。
<nG>	跳至第 n 行。
<U>	撤消（undo）。
<.>	重做（redo）。
<nyy>	拷贝 n 行。
<:e filename>	创建新文件。
<:n filename>	加载新文件。

第二章系统管理

2.1、什么是 init

init 是 Linux 系统操作中不可缺少的程序之一。 是一个由内核启动的用户级进程。内核启动（已经被载入内存，开始运行，并已初始化所有的设备驱动程序和数据结构等）之后，就通过启动一个用户级程序 init 的方式来启动其他用户级的进程或服务。所以,init 始终是第一个进程（其 PID 始终为 1）。
内核会在过去曾使用过 init 的几个地方查找它，它的正确位置（对 Linux 系统来说）是/sbin/init。如果内核找不到 init，它就会试着运行/bin/sh，如果运行失败，系统的启动也会失败。

二、运行级别
运行级就是操作系统当前正在运行的功能级别。这个级别从 1 到 6，具有不同的功能。其功能级别如下：
0 - 停机（千万不能把 initdefault 设置为 0 ）
1 - 单用户模式
2 - 多用户，没有 NFS
3 - 完全多用户模式(标准的运行级)
4 - 没有用到
5 - X11 （xwindow)
6 - 重新启动 （千万不要把 initdefault 设置为 6——把被你黑掉的 linux 的 initdefault 设置为 0 或 6 也算是拒绝服务攻击噢！）
除此之外还有 ABC 三个运行级别，但在 RHLinux 中都没有意义。
这些级别在/etc/inittab 文件里指定。这个文件是 init 程序寻找的主要文件，最先运行的服务是放在/etc/rc.d 目录下的文件。在大多数的 Linux 发行版本中，启动脚本都是位于/etc/rc.d/init.d 中的。这些脚本被用 ln 命令连接到 /etc/rc.d/rcn.d 目录。(这里的 n 就是运行级 0-6)

三、运行级别的配置
运行级别的配置是在/etc/inittab 行内进行的，如下所示：
12 : 2 : wait : / etc / init.d / rc 2
各字段解释如下：
id:runlevels:action:process
id：是一个任意指定的四个字符以内的序列标号，在本文件内必须唯一；使用老版本的 libc5（低于 5.2.18）或 a.out 库编译出来的 sysvinit 限制为 2 字符。注意：像 getty 之类的登陆进

程必须使 id 字段与 tty 编号一致，如 tty1 需要 id=1，许多老版本的登陆进程都遵循这种规则。

runlevels：表示这一行适用于运行那个/些级别（这里是 2，可以有多个，表示在相应的运行级均需要运行）；另外 sysinit、boot、bootwait 这三个进程会忽略这个设置值。

action：表示进入对应的 runlevels 时，init 应该运行 process 字段的命令的方式，常用的字段值及解释在附录内。例子中的 wait 表示需要运行这个进程一次并等待其结束。

process：具体应该执行的命令。例子中的/etc/init.d/rc 命令启动运行级别 2 中应该运行的进程/命令，并负责在退出运行级时将其终止（当然在进入的 runlevel 中仍要运行的程序除外。）

当运行级别改变，并且正在运行的程序并没有在新的运行级别中指定需要运行，那么 init 会先发送一个 SIGTERM 信号终止，然后是 SIGKILL。

有效的 action 值如下：

respawn：表示 init 应该监视这个进程，即使其结束后也应该被重新启动。

wait： init 应该运行这个进程一次，并等待其结束后再进行下一步操作。

once： init 需要运行这个进程一次。

boot： 随系统启动运行，所以 runlevel 值对其无效。

bootwait：随系统启动运行，并且 init 应该等待其结束。

off： 没有任何意义。

initdefault：系统启动后的默认运行级别；由于进入相应的运行级别会激活对应级别的进程，所以对其指定 process 字段没有任何意义。如果 inittab 文件内不存在这一条记录，系统启动时在控制台上询问进入的运行级。

sysinit： 系统启动时准备运行的命令。比如说，这个命令将清除/tmp。可以查看/etc/rc.d/rc.sysinit 脚本了解其运行了那些操作。

powerwait：允许 init 在电源被切断时，关闭系统。当然前提是有 U P S 和监视 U P S 并通知 init 电源已被切断的软件。RH linux 默认没有列出该选项。

powerfail： 同 powerwait，但 init 不会等待正在运行的进程结束。RH linux 默认没有列出该选项。

powerokwait：当电源监视软件报告“电源恢复”时，init 要执行的操作。

powerfailnow：检测到 ups 电源即将耗尽时， init 要执行的操作，和 powerwait/powerfail 不同的哟。

ctrlaltdel：允许 init 在用户于控制台键盘上按下 C t r l + A l t + D e l 组合键时，重新启动系统。注意，如果该系统放在一个公共场所，系统管理员可将 C t r l + A l t + D e l 组合键配置为别的行为，比如忽略等。我是设置成打印一句骂人的话了^o^。

kbrequest：监视到特定的键盘组合键被按下时采取的动作，现在还不完善。

ondemand： A process marked with an ondemand runlevel will be executed whenever the specified ondemand runlevel is called. However, no runlevel change will occur (ondemand runlevels are ‘a’, ‘b’,and ‘c’)

2.2 认识 /etc/passwd 档案与 /etc/shadow 档案

vi /etc/passwd 这个档案的构造是这样的：每一行都代表一个账号，有几行

就代表有几个账号在你的系统中！不过需要特别留意的是，里头很多账号本来就是系统中必须要的（例如 bin, adm, nobody 等等），请不要随意的删除它.

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
wds:x:500:500:wds:/home/wds:/bin/bash
```

上面是系统预设的几个账号，这些账号是系统在使用的！我们先来看一下 root 这个系统管理员这一行好了，你可以明显的看出来，每一行使用:分隔开，共有七个分割，分别是

1.账号名称：对应 UID 用的！例如 root 就是预设的系统管理员的账号名称；

2.密码：早期的 Unix 系统的密码是放在这个档案中的，不过由于这样一来很容易造成数据的被窃取，所以后来就将数据给他改放到 /etc/shadow 中了，这一部份等一下再说，而这里你会看到一个 x ，呵呵！别担心密码已经被移动到 shadow 这个加密过后的档案；

3.UID：这个就是使用者识别码（ID）！通常 Linux 对于 UID 有几个限制需要说给您了解一下：
4.GID：这个与 /etc/group 有关！其实 /etc/group 的观念与 /etc/passwd 差不多，只是他是用来规范 group 的而已！

5.家目录：用户的家目录也可以称为宿主目录，以上面为例， root 的家目录在 /root ，所以当 root 登入之后，马上在的所在就是 /root 里头！呵呵！如果你有个账号的使用空间特别的大，你想要将该账号的家目录移动到其它的硬盘去，没有错！可以在这里进行修改呦！预设的使用者家目录在 /home/yourIDname

6.Shell ：shell 是 Linux 中与 kernl 打交道的一个命令解释器！我们通常使用 /bin/bash 这个 shell 来进行指令的下达！关于 shell 的用法我们会在后面再提及的！这里比较需要注意的是，有一个 shell 可以用来替代成让账号无法登入的指令！那就是 /bin/nologin 这个东西。
shadow 的构造：
由于 /etc/passwd 并不安全，所以后来发展出将密码移动到 /etc/shadow 这个档案中分隔开来的技术！并且加入了很多的限制参数在 /etc/shadow 里头！我们来了解一下这个档案的构造吧！

```
root:$K.K2.hqu.QfV.dkjjteojiasdlkjeo:11661:0:99999:7:::
bin:*:11661:0:99999:7:::
daemon:*:11661:0:99999:7:::
adm:*:11661:0:99999:7:::
```

```
wds:.K1.Hqr23rQfV.dkfasdfasdf1661:0:99999:7:::
```

2.3 系统帐号管理

2.3.1 groupadd

```
[wds @test /root ]# groupadd [-g GID] groupname
参数说明:
-g GID ： 自行设定 GID 的大小
范例:
[wds@localhost ~]# groupadd -g 77 testing<==设定一个群组，GID 为  77
```

这个指令会增加群组！而作用到的档案只有/etc/group 与 /etc/gshadow 这两个档案，说实在的，你也可以直接修改这两个档案就好了，根本不需要使用到这个指令的！使用 vi 修改上面两个档案还比较简单呢！另外，如果你要新增的使用者所要的群组并不存在于系统中，那么您在增加使用者账号之前，就必须要先新增群组！

2.3.2 groupdel

```
[wds @test /root ]# groupdel groupname
参数说明:
范例:
[wds@localhost ~]# groupdel testing
```

这很简单的，就是将 group ID 给他杀掉去！不过，有一点必须要特别留意，就是在杀掉群组之前，请先将该群组的 primary 使用者删除！才好！那什么是 Primary 的使用者呢？说穿了也很简单啦！就是 /etc/passwd 里面，那个 GID 设定为这个群组的 GID 的那个使用者就对啦！

2.3.3 useradd

```
[wds@localhost ~]# useradd [-u UID] [-g GID] [-d HOME] [-mM] [-s shell] username
参数说明:
-u   ： 直接给予一个 UID
-g   ： 直接给予一个 GID （此 GID 必须已经存在于 /etc/group 当中）
-d   ： 直接将他的家目录指向已经存在的目录（系统不会再建立）
```

```
-M   ：不建立家目录
-s   ：定义其使用的 shell
范例：
[wds@localhost ~]# useradd testing    <==直接以预设的数据建立一个名为 testing 的账号
[wds@localhost ~]# useradd -u 720 -g 100 -M -s /bin/bash testing   <==以自己的设定建立账号
```

2.3.4 passwd

```
[wds@localhost ~]# passwd [username]
[wds@localhost ~]# passwd
[wds@localhost ~]# passwd test
Changing password for user test
New password:     <==输入密码
BAD PASSWORD: it is based on a dictionary word
Retype new password:   <==再输入一次！
passwd: all authentication tokens updated successfully
```

这个指令可以修改使用者的密码！要注意的是，这个指令在 /bin/passwd 中，而账号所存放的地方在 /etc/passwd 中. 一般使用者的用法是直接输入 passwd 即可； root 可以使用 passwd [username] 来替 username 这个账号取一个新的密码

2.3.5 su

```
[wds@localhost ~]# su
参数说明：
范例：
[wds@localhost ~]# su
Password:          <==输入 root 的密码
[wds@localhost ~]#       <==身份变成 root 了！
[wds@localhost ~]# su -  <==连环境参数档案都是读取 root 的！
[wds@localhost ~]# su test  <==将 root 的身份改为 test ，且不需要输入密码！
```

这个指令很有用,这是将普通拥护转换成管理员的指令！通常为了安全的考虑， telnet 与 ssh 尽量不要以 root 的身份来登入！但是有时后我们又要在外头以 root 的身份来修改系统设定，这个时候 su 就很有用了！su 的使用真的很简单，输入 su 之后，直接给他输入 root 的密码，此时您就是 root 了！但是需要特别留意的是：

2.3.6 w 查看当前登陆用户

```
[wds@localhost ~]# w
00:17:54 up 48 days, 23:28,  1 user,  load average: 0.12, 0.16, 0.17
```

```
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
wangdong pts/0    58.155.233.6      00:17    0.00s  0.02s  0.00s w
```

说明:

本命令主要是用来查看登陆用户名，IP，时间，等一些信息

2.3.7 ps 查看进程

```
[wds@localhost ~]# ps  -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 Jun10 ?        00:00:00 init [3]
root         2     1  0 Jun10 ?        00:00:00 [migration/0]
root         3     1  0 Jun10 ?        00:00:00 [ksoftirqd/0]
root         4     1  0 Jun10 ?        00:00:00 [watchdog/0]
root         5     1  0 Jun10 ?        00:00:00 [events/0]
root         6     1  0 Jun10 ?        00:00:00 [khelper]
root         7     1  0 Jun10 ?        00:00:00 [kthread]
root        10     7  0 Jun10 ?        00:00:01 [kblockd/0]
root        11     7  0 Jun10 ?        00:00:00 [kacpid]
root        89     7  0 Jun10 ?        00:00:00 [cqueue/0]
root        92     7  0 Jun10 ?        00:00:00 [khubd]
root        94     7  0 Jun10 ?        00:00:00 [kseriod]
root       153     7  0 Jun10 ?        00:00:00 [pdflush]
root       154     7  0 Jun10 ?        00:00:00 [pdflush]
```

本命令主要查看本机器的详细进程情况.

2.3.8 kill 命令

```
[wds@localhost ~]# ps  -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root       154     7  0 Jun10 ?        00:00:00 [pdflush]
[wds @test /root]# kill 154
[wds @test /root]# kill -9 154
[wds @test /root]# kill  -HUP 154
```

2.3.9 ifocnfig 查看网络命令

```
[wds@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0A:EB:23:8B:90
          inet addr:192.168.102.131  Bcast:192.168.102.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2409578 errors:0 dropped:0 overruns:0 frame:0
          TX packets:965998 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:840272734 (801.3 MiB)  TX bytes:918115554 (875.5 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2307728 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2307728 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:161369570 (153.8 MiB)  TX bytes:161369570 (153.8 MiB)
```

- inet addr:192.168.102.131 本机的 IP 地址
- Bcast:192.168.102.255 本机的广播地址:用于同时发信息给网络上的其他地址
- Mask:255.255.255.0 本机淹码:本机与网络其他机器进行通信时，判断是否在同一网内.
- RX bytes:840272734 (801.3 MiB) RX 接收数据大小
- TX bytes:918115554 (875.5 MiB) TX 发送数据大小
- MTU:每个数据包的最大传输单元位(MAXimum Transmission Unit)用来控制数据包大小，默认是 1500

Ifconfig 除用于查看本机的 IP 地址以外还可以修改本记的 IP 地址和关闭网卡和开网卡等功能.

```
[wds@localhost ~]# ifconfig eth0 192.168.102.10 netmask 25.255.255.0
eth0      Link encap:Ethernet  HWaddr 00:0A:EB:23:8B:90
          inet addr:192.168.102.10  Bcast:192.168.102.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2409578 errors:0 dropped:0 overruns:0 frame:0
          TX packets:965998 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:840272734 (801.3 MiB)  TX bytes:918115554 (875.5 MiB)

[wds @test /root]# ifconfig eth0 down
[wds @test /root]# ifconfig eth0 up
```

2.3.10 查看网络状态命令

```
[wds@localhost ~]# netstat  -an | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 0.0.0.0:3306                0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:10000               0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:21                  0.0.0.0:*                   LISTEN
tcp        0      0 :::80                       :::*                        LISTEN
tcp        0      0 :::22                       :::*                        LISTEN
tcp        0     52 ::ffff:192.168.102.131:22   ::ffff:58.155.233.6:1240
ESTABLISHED
udp        0      0 0.0.0.0:10000               0.0.0.0:*
udp        0      0 127.0.0.1:7288              127.0.0.1:53
ESTABLISHED
```

本命令查看本机的开放端口与网络状态等

2.3.11 lsof 查看端口在做什么

```
[wds@localhost ~]# netstat  -an | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 0.0.0.0:3306                0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:10000               0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:21                  0.0.0.0:*                   LISTEN
tcp        0      0 :::80                       :::*                        LISTEN
tcp        0      0 :::22                       :::*                        LISTEN

[wds @test /root]# lsof -i:80
COMMAND   PID USER   FD   TYPE   DEVICE SIZE NODE NAME
httpd   28126 root    3u  IPv6 87902392       TCP *:http (LISTEN)
httpd   28141  www    3u  IPv6 87902392       TCP *:http (LISTEN)
httpd   28142  www    3u  IPv6 87902392       TCP *:http (LISTEN)
httpd   28143  www    3u  IPv6 87902392       TCP *:http (LISTEN)
httpd   28144  www    3u  IPv6 87902392       TCP *:http (LISTEN)
httpd   28145  www    3u  IPv6 87902392       TCP *:http (LISTEN)
httpd   28158  www    3u  IPv6 87902392       TCP *:http (LISTEN)
httpd   29682  www    3u  IPv6 87902392       TCP *:http (LISTEN)
```

第三章Raid 制作方法

3.1 raid 的简介

安装程序实现软件 RAID 代替硬件 RAID 的方法，今天再进一步谈谈手动创建软 RAID 和日常维护的方法。mdadm 使用的也是 md 驱动，由于其拥有多种模式，而且单一工具，不依赖任何配置文件，是替代 raidtools 的好工具。目前几乎所有发行版本使用的都是该工具.

源码下载：http://www.cse.unsw.edu.au/~neilb/source/mdadm/

3.2.安装和编译

[wds@localhost]# tar xzvf ./mdadm-1.6.0.tgz
[wds@localhost]#cd mdadm-1.6.0
[wds@localhost]# make install

[wds@localhost]#rpm -ivh mdadm-1.6.0-3.rpm

3.3 磁盘分区

只能使用 Sofware RAID 格式的磁盘才能组成阵列，所以，首先我们要把做好磁盘格式。
正如上面提到的，除了系统盘 sda 外，我们需要对 sdb、sdc、sdd 进行操作

fdisk /dev/sdb

```
[root@localhost ~]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklab
el
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
```

执行 p 分区前状态：

```
Command (m for help): p

Disk /dev/sdb: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
```

n，划分区：

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-130, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-130, default 130):
Using default value 130
```

t，修改分区格式为 fd：

```
Command (m for help): t
Selected partition 1
Hex code (type L to list codes): fd
Changed system type of partition 1 to fd (Linux raid autodetect)
```

w，保存：

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

b）同样的方法，对 sdc、sdd 进行分区和保存。

```
Disk /dev/sdb: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         130     1044193+  fd  Linux raid autodetect

Disk /dev/sdc: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1               1         130     1044193+  fd  Linux raid autodetect

Disk /dev/sdd: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdd1               1         130     1044193+  fd  Linux raid autodetect
```

然后需要从起计算机

3.4 创建磁盘阵列

mdadm 可以支持 LINEAR、RAID0 (striping)、 RAID1(mirroring)、 RAID4、RAID5、RAID6 和 MULTIPATH 的阵列模式。

创建命令格式如下：

mdadm [mode] <raiddevice> [options] <component disks>

例如：创建一个 RAID 0 设备：

mdadm --create --verbose /dev/md0 --level=0 --raid-devices=3 /dev/sdb1 /dev/sdc1 /dev/sdd1

sdb1 /dev/sdc1 /dev/sdd1

--level 表示创建的阵列模式，--raid-devices 表示参与阵列的磁盘数量。

```
[root@localhost ~]# mdadm --create --verbose /dev/md0 --level=0 --raid-devices=3
 /dev/sdb1 /dev/sdc1 /dev/sdd1
mdadm: chunk size defaults to 64K
mdadm: array /dev/md0 started.
```

也可以这样表达，意思是一样的：

mdadm -Cv /dev/md0 -l0 -n3 /dev/sd[bcd]1

还可以增加-c128 参数，指定 chunk size 为 128K（默认 64K）

3.5 修改磁盘阵列配置

mdadm 不采用/etc/mdadm.conf 作为主要配置文件，它可以完全不依赖该文件而不会影响阵列的正常工作。该配置文件的主要作用是方便跟踪软 RAID 的配置。对该配置文件进行配置是有好处的，但不是必须的。推荐对该文件进行配置。通常可以这样来建立：

```
echo DEVICE /dev/sd[bcd]1 > /etc/mdadm.conf
mdadm -Ds >> /etc/mdadm.conf
mdadm --detail --scan >> /etc/mdadm.conf
```

3.6、格式化阵列

后续，只要你把/dev/md0 作为一个单独的设备来进行操作即可：

mkfs.ext3 /dev/md0

mkdir /mnt/test
mount /dev/md0 /mnt/test

3.7 开机自动加载

若要开机自动挂载，请加入/etc/fstab 中：

/dev/md0 /mnt/tes auto defaults 0 0

第四章 NFS 简介

4.1 NFS 简介

NFS 为 network file system 的简称，最早由 sun 公司开发，一般 NFS 广泛应用在集群服务器上，他的最大特点是可以通过网络让不同的机器，不同的操作系统可以彼此的共享文件，所以它可以看作一个简单的文件服务器。NFS 其实可以被视为一个 RPC 服务程序，在启动 RPC 程序前我们先要做好端口的映射工作这就是 portmap,portmap 的意思是当 Client 要连接服务器时必须知道服务器的一个空闲端口这时 Client 会向服务器的 portmap 请求一个端口然，然后 Server 告诉 Client 这端口后才可以建立连接，所以在启动 NFS 前要先启动 portmap

4.2 编译 nfs

[wds@localhost ~]# rpm –qa |grep nfs && rpm –qa | grep portmap #查找这两个是否安装

[wds@localhost ~]# vi /etc/exports　　# 这文件是 NFS 的主要配置文件
[wds@localhost ~]# /usr/sbin/exportfs　#这个文件是 nfs 共享资源命令
[wds@localhost ~]# /usr/sbin/showmount　#可以查看远程服务器的共享目录
[wds@localhost ~]# /var/lib/nfs/xtab　#nfs 的日志文件

4.3 NFS 配置

[wds@localhost ~]# vi /etc/exports
[你想要的共享的目录] + ip 地址(参数一，参数二) [主机名二]（参数三，参数四）
参数列表
rw:　　可以写入权限
ro:　　只读权限
no_root_squash:　　登陆 NFS 主机共享目录的如果是 root 用户那么那的权限也为 root 但是这样并不安全
root_squash:　　登陆的用户如果为 root 它的权限将变成 nobody
all_squash:　　不论登陆的用户是什么用户都以匿名用户的权限
sync:　　数据同步写入硬盘和内存中
async:　　数据先暂时存放在内存中，而不写入硬盘
anounid:　　这个可以自己设定 uid,但是必须与/etc/passwd 目录中用户 uid 一样
anongid:　　同 anonuid,但是变的是 group id

4.4 服务器端配置

[wds@localhost ~]# service　portmap start　　#首先打开 portmap
[wds@localhost ~]# service　nfs　start　　# 在打开 nfs
[wds@localhost ~]# iptables –F　　#清空防火墙命令
[wds@localhost ~]#
比如说我要共享/var/www/html 目录 但是只是让和我一个网段的机器访问 192.168.0.0/24 这个网段读或写，其他的就只能读，然后在发布一个私人目录/home/wds/只开放给 192.168.0.8 这个 IP
[wds@localhost ~]# vi /etc/exports
/var/www/html　　192.168.0.0/24 (rw)　　*(ro)
/home/wds　　192.168.0.8(rw)

现在想要*.chinaunix.com 网段的机器登陆我的 NFS，并且访问我的/home/wds/ 但是它们存储时我希望它们的 uid 和 gid 都变成 40 这个用户身份
[wds@localhost ~]# vi /etc/exports
/var/www/html　　192.168.0.0/24 (rw)　　*(ro)
/home/wds　　192.168.0.8(rw)
/home/wds　　*.chinaunix.com(rw,all)squash,anounid=40,anongid=40)

如果我们修改/etc/exports 这个文件后，是否要从新启动 nfs 呢？答案是不不需要，只要使用 exportfs 来从新扫描一次/etc/exports 文件,并且从新设置文件加载即可

语法为：
[wds@localhost ~]# exportfs [-aruv]
参数说明：
-a: 全部挂载（或者卸载）/etc/exports 文件的设置
-r: 从新挂载/etc/exports 里设置，此外，同步更新/etc/exports 及/var/lib/nfs/xtab 的内容
-u: 卸载某一目录
-v: 在导出时，将共享目录显示在屏幕上
例如:
[wds@localhost ~]# exportfs –rv 全部从新导出一次
[wds@localhost ~]# exportfs –au 全部卸载掉

Showmount 的是显示是否有挂载

语法为：
[wds@localhost ~]# showmount [-ae] hostname
参数说明：
-a: 在屏幕上显示与当前的 client 连接后使用目录的状态
-e: 显示 Hostname 这台机器的/etc/exports 中的共享信息
[wds@localhost log]# showmount -e localhost
Export list for localhost:
/var/www/html (everyone)

Rpcinfo [-p]hostname[or ip]
-p 显示端口与程序的信息

[wds@localhost log]# rpcinfo -p localhost

program	vers	proto	port	
100000	2	tcp	111	portmapper
100000	2	udp	111	portmapper
100024	1	udp	1024	status
100024	1	tcp	1024	status
100011	1	udp	837	rquotad
100011	2	udp	837	rquotad
100011	1	tcp	840	rquotad
100011	2	tcp	840	rquotad
100003	2	udp	2049	nfs
100003	3	udp	2049	nfs
100003	4	udp	2049	nfs
100003	2	tcp	2049	nfs
100003	3	tcp	2049	nfs

```
100003      4     tcp     2049    nfs
100021      1     udp      1026    nlockmgr
100021      3     udp      1026    nlockmgr
100021      4     udp      1026    nlockmgr
100021      1     tcp     1026    nlockmgr
100021      3     tcp     1026    nlockmgr
100021      4     tcp     1026    nlockmgr
100005      1     udp      858    mountd
100005      1     tcp      861    mountd
100005      2     udp      858    mountd
100005      2     tcp      861    mountd
100005      3     udp      858    mountd
100005      3     tcp      861    mountd
```

4.5 Client 端的设置

Server 端设置完毕,接下来就是让 client 端连接上 server!连接 server 步骤如下:

1. 扫描可以使用的 server 目录:
2. 在 client 端建立装载点
3. 使用 mount 命令远程挂载远程共享目录
4. 解决可能发生的问题（被防火墙过滤掉了）

Showmount 是显示远程主机共享资源

```
[wds@localhost ~]# showmount -e 192.168.0.8
Export list for 192.168.0.8:
/var/www/html (everyone)
/home/wds       *.chinaunix.com,192.168.0.6
```

[wds@localhost ~]# mount -t nfs 192.168.0.8:/var/www/html /mnt 把远程的 /var/www/html 挂载到本地

```
[wds@localhost ~]# umount /mnt          卸载远程目录
```

如果你想要开机启动时自动加载 NFS 服务器导出目录，我们在 NFS 端/etc/fstab 文件中加入以下一行

```
192.168.0.8:/var/www/html    /mnt    nfs    rsize=8192,wsize=8192,timeo=14,intr
```

第五章 vsftpd 配置

5.1 vsftpd 简介

vsftpd 是目前各 linux 发行版比较常见的 FTP 软件,vsftpd 全名为[very secure ftp daemon],意思是非常安全的 ftp 服务器的意思，官方网站 http://vsftpeasts.org/。

5.2 编译安装 vsftpd

[wds@localhost ~]# yum install vsfptd (centos)
[wds@localhost ~]# apt-get install vsftpd (ubuntu)

[wds@localhost ~]# tar -zxvf vsftpd-2.0.5.tar.gz (源码编译)
[wds@localhost ~]# cd vsftpd-2.0.5.

[wds@localhost ~]# make
[wds@localhost ~]# make install

5.2 vsftpd 文件结构

/etc/vsftpd.ftpusers 里面的使用者账号均无法使用 vsftpd ！
/etc/vsftp.user_list：且与实体用户有关！当我们在 vsftpd.conf 里面设定好了实体用户的使用者没有被 chroot 到自己的家目录下(也就是使用者登入后不只能到自己的家目录，还可以跳到其它目录)，不过，某些使用者您想让他无法离开家目录时，预设在 /etc/vsftp.user_list 这个档案里面，就可以将该使用者限制在自己的家目录内了！一行一个账号。

/usr/local/sbin/vsftpd 或 /usr/sbin/vsftpd：这就是 vsftpd 的主要执行档咯！不要怀疑， vsftpd 只有这一个执行档而已啊！

/var/ftp：这个是 vsftpd 的预设匿名者登入的根目录！

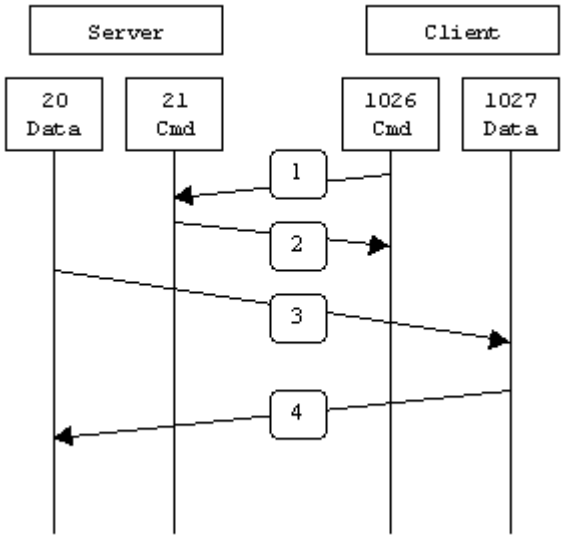
5.4 主动模式与被动模式

主动模式 FTP

主动模式的 FTP 是这样的：客户端从一个任意的非系统保留端口 N（N>;1024）连接到 FTP 服务器的命令端口，也就是 21 端口。然后客户端开始监听 端口 N+1，并发送 FTP 命令“port N+1”到 FTP 服务器。接着服务器会从它自己的数据端口（20）连接到客户端指定的数据端口（N+1）。

针对 FTP 服务器前面的防火墙来说，必须允许以下通讯才能支持主动方式 FTP：

1.任何端口到 FTP 服务器的21端口 （客户端初始化的连接 S<-C）
2.FTP 服务器的21端口到大于1023的端口（服务器响应客户端的控制端口 S->C）
3.FTP 服务器的20端口到大于1023的端口（服务器端初始化数据连接到客户端的数据端口 .S->C）
4.大于1023端口到 FTP 服务器的20端口（客户端发送 ACK 响应到服务器的数据端口 S<-C）

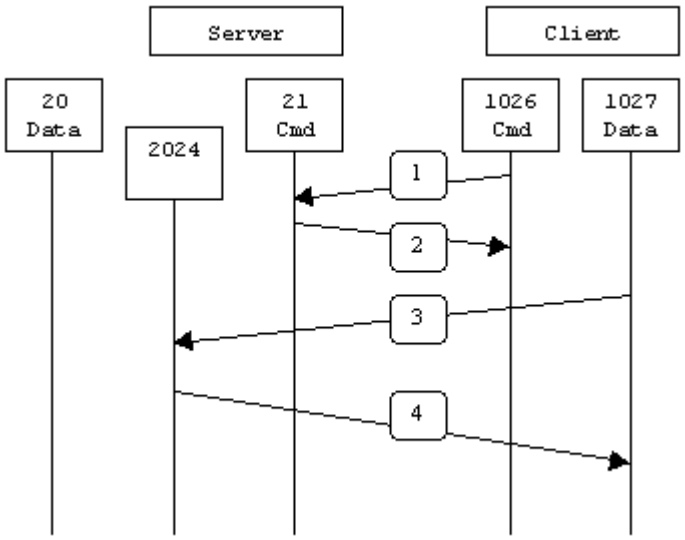


被动模式 FTP

为了解决服务器发起到客户的连接的问题，人们开发了一种不同的 FTP 连接方式。这就是所谓的被动模式，或者叫做 PASV，当客户端通知服务器它处于被动模 式时才启用。

在被动方式 FTP 中，命令连接和数据连接都由客户端，这样就可以解决从服务器到客户端的数据端口的入方向连接被防火墙过滤掉的问题。当开启一个 FTP 连接 时，客户端打开两个任意的非系统保留端口（N >; 1024 和 N+1）。第一个端口连接服务器的 21 端口，但与主动方式的 FTP 不同，客户端不会提交 PORT 命令并允许服务器来回连它的数据端口，而是提交 PASV 命令。这样做的结果是服务器会开启一个任意的非特权端口（P >; 1024），并发送 PORT P 命令给客户端。然后客户端发起从本地端口 N+1 到服务器的端口 P 的连接用来传送数据。

对于服务器端的防火墙来说，必须允许下面的通讯才能支持被动方式的 FTP:
1.从任何端口到服务器的21端口 （客户端初始化的连接 S<-C）
2.服务器的21端口到任何大于1023的端口 （服务器响应到客户端的控制端口的连接 S->C）
3.从任何端口到服务器的大于1023端口 （入；客户端初始化数据连接到服务器指定的任意端口 S<-C）
4.服务器的大于1023端口到远程的大于1023的端口（出；服务器发送 ACK 响应和数据到客户端的数据端口 S->C）



5.5 vsftpd.conf 文件介绍

connect_from_port_20=YES | NO
这个设定项目在启动主动联机的 port 20 ！

listen_port=21
默认 ftp 端口

dirmessage_enable=YES | NO
当使用者进入某个目录时，会显示该目录需要注意的内容，显示的档案预设是 .message ，当然，可以使用底下的设定项目来修订！

message_file=.message
当 dirmessage_enable=YES 时，可以设定这个项目来让 vsftpd 寻找该档案来显示讯息！
listen=YES | NO
若设定为 YES 表示 vsftpd 是以 standalone 的方式来启动的！

pasv_enable=YES | NO
启动被动式联机(passive mode)，建议设置 YES！

use_localtime=YES | NO
是否使用主机的时间？！预设使用 GMT 时间(格林威治).

write_enable=YES | NO
是否允许使用者具有写入的权限.

connect_timeout=60
单位是秒，如果 client 尝试连接我们的 vsftpd 命令通道超过 60 秒，则不等待，强制断线。
accept_timeout=60
当使用者以被动式 PASV 来进行数据传输时，如果主机启用 passive port
并等待 client 超过 60 秒，那么就给他强制断线！您可以修改 60 这个数值。

data_connection_timeout=300
如果 client 与 Server 间的数据传送在 300 秒内都无法传送成功，那 Client 的联机就会被我们的 vsftpd 强制剔除！

idle_session_timeout=300
如果使用者在 300 秒内都没有命令动作，强制离线！

max_clients=0
如果 vsftpd 是以 stand alone 方式启动的，那么这个设定项目可以设定 同一时间，最多有多少 client 可以同时连上 vsftpd ？

max_per_ip=0
与上面 max_clients 类似，这里是同一个 IP 同一时间可允许多少联机？

pasv_max_port=0
pasv_min_port=0
上面两个是与 passive mode 使用的 port number 有关，如果您想要使用
65400 到 65410 这 11 个 port 来进行被动式资料的连接，可以这样设定
pasv_max_port=65410 以及 pasv_min_port=65400

ftpd_banner=一些文字说明
当使用者无法顺利连上我们的主机，例如联机数量已经超过 max_clients
的设定了，那么 client 的画面就会显示『一些文字说明』的字样，您可以修改
关于实体用户登入者的设定值

guest_enable=YES | NO
若这个值设定为 YES 时，那么任何非 anonymous 登入的账号，均会被 假设成为 guest (访客) ！

local_enable=YES | NO
这个设定值必须要为 YES 时，在 /etc/passwd 内的账号才能以实体用户的方式登入我们的 vsftpd 主机！

local_max_rate=0
实体用户的传输速度限制，单位为 bytes/second， 0 为不限制。

chroot_local_user=YES |NO
将使用者限制在自己的家目录之内(chroot)！这个设定在 vsftpd 当中预设是 NO，因为有底

下两个设定项目的辅助！ 所以不需要启动他！

chroot_list_enable=YES | NO
是否启用将某些实体用户限制在他们的家目录内？！预设是 NO ， 不过，如果您想要让某些使用者无法离开他们的家目录时， 可以考虑将这个设定为 YES ，并且规划下个设定值 chroot_list_file=/etc/vsftpd.chroot_list
如果 chroot_list_enable=YES 那么就可以设定这个项目了！他里面可以规定 那一个实体用户会被限制在自己的家目录内而无法离开！(chroot) 一行一个账号即可！

userlist_deny=YES (NO)
若此设定值为 YES 时，则当使用者账号被列入到某个档案时，在该档案内
的使用者将无法登入 vsftpd 服务器！该档案文件名与下列设定项目有关。

userlist_file=/etc/vsftpd.user_list
若上面 userlist_deny=YES 时，则这个档案就有用处了！在这个档案内的 账号都无法使用 vsftpd ！

anonymous_enable=YES | NO
设定为允许 anonymous 登入我们的 vsftpd 主机！预设是 YES ，底下的所有 相关设定都需要将这个设定为 anonymous_enable=YES 之后才会生效！

anon_world_readable_only=YES | NO
仅允许 anonymous 具有下载可读档案的权限，预设是 YES。

anon_other_write_enable=YES | NO
是否允许 anonymous 具有写入的权限？预设是 NO！如果要设定为 YES， 那么开放给 anonymous 写入的目录亦需要调整权限，让 vsftpd 的 PID 拥有者可以写入才行！

anon_mkdir_write_enable=YES | NO
是否让 anonymous 具有建立目录的权限？默认值是 NO！如果要设定为 YES，
那么 anony_other_write_enable 必须设定为 YES ！

anon_upload_enable=YES |NO
是否让 anonymous 具有上传数据的功能，预设是 NO，如果要设定为 YES 则
anon_other_write_enable=YES 必须设定。

deny_email_enable=YES | NO
将某些特殊的 email address 抵挡住，不让那些 anonymous 登入！
如果以 anonymous 登入主机时，不是会要求输入密码吗？密码不是要您
输入您的 email address 吗？如果你很讨厌某些 email address ，
就可以使用这个设定来将他取消登入的权限！需与下个设定项目配合：

banned_email_file=/etc/vsftpd.banned_emails
如果 deny_email_enable=YES 时，可以利用这个设定项目来规定那个

email address 不可登入我们的 vsftpd 喔！在上面设定的档案内，
一行输入一个 email address 即可！

ano_anon_password=YES |NO
当设定为 YES 时，表示 anonymous 将会略过密码检验步骤，
而直接进入 vsftpd 服务器内！所以一般预设都是 NO 的！

anon_max_rate=0
这个设定值后面接的数值单位为 bytes/秒 ，限制 anonymous 的传输速度，
如果是 0 则不限制(由最大频宽所限制)，如果您想让 anonymous 仅有
30 KB/s 的速度，可以设定『anon_max_rate=30000』

anon_umask=077
限制 anonymous 的权限！如果是 077 则 anonymous 传送过来的档案
权限会是 -rw------- ！

ascii_download_enable=YES |NO
如果设定为 YES ，那么 client 就可以使用 ASCII 格式下载档案。
一般来说，由于启动了这个设定项目可能会导致 DoS 的攻击，因此预设是 NO。

ascii_upload_enable=YES |NO
与上一个设定类似的，只是这个设定针对上传而言！预设是 NO。

async_abor_enable=YES |NO
如果您的 FTP client 会下达 "async ABOR" 这个指令时，这个设定才需要启用
一般来说，由于这个设定并不安全，所以通常都是将他取消的！

check_shell=YES (NO)
  如果您想让拥有任何奇怪的 shell 的使用者(在 /etc/passwd 的 shell 字段)
  可以使用 vsftpd 的话，这个设定可以设定为 NO 喔！
aone_process_model=YES (NO)
这个设定项目比较危险一点～当设定为 YES 时，表示每个建立的联机
都会拥有一支 process 在负责，可以增加 vsftpd 的效能。不过，
除非您的系统比较安全，而且硬件配备比较高，否则容易耗尽系统资源喔！
一般建议设定为 NO！

tcp_wrappers=YES |NO
当然我们都习惯支持 TCP Wrappers 的啦！所以设定为 YES 吧！

xferlog_enable=YES |NO
当设定为 YES 时，使用者上传与下载档案都会被纪录起来。记录档案
与下一个设定项目有关：

xferlog_file=/var/log/vsftpd.log

如果上一个 xferlog_enable=YES 的话，这里就可以设定了！这个是登录档的档名啦！

xferlog_std_format=YES |NO
是否设定为 wu ftp 相同的登录档格式？！预设为 NO ，因为登录档会比较容易读！
不过，如果您有使用 wu ftp 登录文件的分析软件，这里才需要设定为 YES

nopriv_user=nobody
我们的 vsftpd 预设以 nobody 作为此一服务执行者的权限。因为 nobody 的权限
相当的低，因此即使被入侵，入侵者仅能取得 nobody 的权限喔！

pam_service_name=vsftpd
这个是 pam 模块的名称，我们放置在 /etc/pam.d/vsftpd

5.6 Vsftpd 虚拟用户配置文档

5.6.1 基于文件 db

1) 创建虚拟用户的数据库
我们将使用 pam_user.db 来认证虚拟用户。这需要提供一个“db"格式（一种通用数据库格式,如果没有请 yum -y install db4-utils）的用户名/密码文件。
创建一个"db"格式的文件，首先要创建一个标准文本文件，并把用户名，密码以竖直排列方式输入。如 vi logins.txt.

admin 管理员用户 最高权限
wds 密码

wds 管理员用户
123 密码

sunbird 一般用户
123

以 ROOT 登录，创建一个数据库文件，如下：
db_load -T -t hash -f logins.txt /etc/vsftpd_login.db
（这要求 berkeley db 程序已经安装）
(注：一些系统也许安装了多个版本的"db",所以某些情况下你可能使用"db3_load"才是正确的。关键在于要让 pam_userdb 相信它的登录数据库是哪一个 db 版本所产生（一般都是 db3,尽管你的系统里可能安装的是 db4).）

这将创建/etc/vsftpd_login.db 文件。显然，你希望设定这个文件的权限：

chmod 600 /etc/vsftpd_login.db

2）用你的新数据库创建一个 PAM 文件
请参考范例 vsftpd.pam，它包含 2 行：
auth required /lib/security/pam_userdb.so db=/etc/vsftpd_login
account required /lib/security/pam_userdb.so db=/etc/vsftpd_login

这是告诉 PAM 用新的数据库去验证用户。把这个 PAM 文件拷贝到 PAM 目录，一般是 /etc/pam.d
cp vsftpd.pam /etc/pam.d/ftp

3)为虚拟用户设置 home 目录
useradd -d /home/ftp virtual
ls -ld /home/ftp
(which should give):
drwx------ 3 virtual virtual 4096 Jul 30 00:39 /home/ftpsite

我们已经创建了一个名叫"virtual"的用户，home 目录是"/home/ftpsite".
我们拷贝一些东西到这个下载目录：
cp /etc/hosts /home/ftp
chown virtual.virtual /home/ftp/hosts

4)配置/etc/vsftpd/vsftpd.conf

vsftpd.conf
anonymous_enable=NO
local_enable=YES
write_enable=NO
anon_upload_enable=NO
anon_mkdir_write_enable=NO
anon_other_write_enable=NO
chroot_local_user=YES
guest_enable=YES
guest_username=virtual
listen=YES
listen_port=21
pasv_min_port=30000
pasv_max_port=3099

1）激活单个用户配置功能。
要激活这个功能，需要增加以下配置行到配置文件：

user_config_dir=/etc/vsftpd_user_conf

并且创建目录：

mkdir /etc/vsftpd_user_conf

用户配置
vi /etc/vsftpd_user_conf/admin　　可读，可写，可改名
anon_world_readable_only=NO
write_enable=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES

vi /etc/vsftpd_user_conf/sunbird　　只能读
anon_world_readable_only=NO

一个特殊功能的实现，当你登陆到 FTP 上后你看不见文件里的内容，但是你如果你知道 FTP 里的文件名你可以直接下载，这是为了防止某些人直接上到 FTP 上去下载所有文件。

vi /etc/vsftpd/vsftpd.conf

anonymous_enable=NO
local_enable=YES
write_enable=NO
anon_upload_enable=NO
anon_mkdir_write_enable=YES　　该成 yes
anon_other_write_enable=YES　　该成 yes
chroot_local_user=YES
guest_enable=YES
guest_username=virtual
listen=YES
listen_port=21
pasv_min_port=30000
pasv_max_port=30999
user_config_dir=/etc/vsftpd_user_conf

vi /etc/vsftpd_user_conf/wds　　只能读 而且看不见
anon_world_readable_only=NO

如果想更改目录的话
anon_world_readable_only=NO
local_root=/home/sunbird/

5.6.2 基于 mysql 的虚拟用户

1) 手下下载 pam_mysql (http://sourceforge.net/projects/pam- mysql/或本站下载)

2) 安装 pam_mysql-

[wds@localhost ~]# tar -xvzf pam_mysql-0.7RC1.tar.gz
[wds@localhost ~]# cd pam_mysql-0.7RC1/
[wds@localhost ~]# ./configure --with-mysql= mysql 安装路径
[wds@localhost ~]# make
[wds@localhost ~]# make install
[wds@localhost ~]#cp .libs/pam_mysql.so /lib/security/pam_mysql.so

[wds@localhost ~]#vim /etc/pam.c/ftp 加入以下内容

```
auth required /lib/security/pam_mysql.so user=virtual passwd=1q2w3e host=localhost db=vsftp table=user usercolumn=name passwdcolumn=passcrypt=0

account required /lib/security/pam_mysql.so user=virtual passwd=1q2w3e host=localhost db=vsftp table=user usercolumn=name passwdcolumn=pass crypt=0
```

相关字段对应含义
Pam_mysql 认证路径 , user=登录数据库用户名 , passwd=数据库密码
Host= 本地 , db =数据库名 , table=数据库表名 ,
usercolumn=表结构中的登录 ftp 用户名
passwdcolumn=表结构中的登录 ftp 密码
crypt=0 为明文密码
crypt=1 为 md5 密码
crypt=2 为 mysql 中 password 密码

3) mysql

建立表
mysql> create databases vsfptd

```
mysql> create    table    user (
mysql> `name` varchar(20)    not null,
mysql> `pass` varchar(20)    not null,
mysql>    primary key(name));
```

插入测试数据

```
mysql> insert into user (name,pass) value ('test1',"test1");
```

建立用户

```
mysql> grant all privileges on vsftpd.* to virtual@localhost identified by '1q2w3e';
mysql>flush privileges;
```

4). 建立用户 useradd virtual

5). 编辑 vsftpd 主配置文件

```
anonymous_enable=NO
local_enable=YES
write_enable=NO
anon_upload_enable=NO
anon_mkdir_write_enable=NO
anon_other_write_enable=NO
chroot_local_user=YES
guest_enable=YES
guest_username=virtual
listen=YES
listen_port=21
pasv_min_port=30000
pasv_max_port=3099
user_config_dir=/etc/vsftpd_user_conf
xferlog_enable=YES
```

6) 建立用户配置文件目录 mkdir /etc/vsftpd_user_conf

7) 编辑用户权限

vim /etc/vsftpd_user_conf/test1 插入以下内容

```
anon_world_readable_only=NO
write_enable=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES
```

8) 重启 vsftpd 进程，并测试.

5.7 ftps 配置

ftp 登录时是以明文形式登录的，所以非常不安全，通过 ftps 形式可以加密传输数据，这样对一些安全度要求比较高的行业比较适用，以下为配置 ftps 详细过程。

1) 配置证书

```
[wds@localhost ~]#    mkdir /etc/vsftpd/.ssl
[wds@localhost ~]#    cd /etc/vsftpd/.ssl
```

```
[wds@localhost ~]#    openssl req -new -x509 -nodes -out vsftpd.pem -keyout vsftpd.pem

Generating a 1024 bit RSA private key
.....................++++++
.............++++++
writing new private key to 'vsftpd.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:CN                国家
State or Province Name (full name) [Berkshire]:beijing              所在省
Locality Name (eg, city) [Newbury]:beijing                  所在城市
Organization Name (eg, company) [My Company Ltd]:365online.me      公司
Organizational Unit Name (eg, section) []:365online.me        组织名
Common Name (eg, your name or your server's hostname) []:365online.me      域名
Email Address []:7717060@sina.com      邮箱地址
```

2) 在 vsftpd.conf 中加入以下内容

```
ssl_enable=YES                支持 ssl
ssl_sslv2=YES                 支持安全套接字 v2
ssl_sslv3=YES                 支持安全套接字 v3
```

ssl_tlsv1=YES 支持支持 tls 加密方式 v1
force_local_logins_ssl=YES 强制非匿名用户使用加密方式登录和传输数据，如果为 NO,
force_local_data_ssl=YES 可以选择加密和不加密
rsa_cert_file=/etc/vsftpd/.ssl/vsftpd.pem 证书位置.

3) 通过 cuteftp 软件来测试 ftps 是否可用

5.8 某学院 vsftpd 应用案例

5.8.1 应用案例介绍

1) 某学院门户网站向学院内部学生提供电影下载服务，建立 ftp 需求 1.保证安全 2. 不同帐户设置不同权限如 admin 管理员可以上传电影和修改电影名,guest 只能下载 ftp 中内容且看不见 ftp 中都有什么东西,防止用户批量下载,3. 只准许 192.168.1.0/24 访问 ftp 服务器。

2) vsftpd 的安装
[wds@localhost ~]# yum install vsftpd*
[wds@localhost ~]# yum install db4-utils*

3) 设置虚拟用户 login.txt
[wds@localhost ~]# vim login.txt 将以下内容加入 login.txt
admin
123qweasdzxc
guest
guest

4) 生成 db 文件
[wds@localhost ~]# db_load -T -t hash -f login.txt /etc/vsftpd_login.db

5) vim /etc/pam.d/ftp 将以下内容写入 ftp
auth required /lib/security/pam_userdb.so db=/etc/vsftpd_login
account required /lib/security/pam_userdb.so db=/etc/vsftpd_login

6) 给系统建立虚拟账号权限 useradd virtual

7) vim /etc/vsftpd/vsftpd.conf 将以下内容写入
anonymous_enable=NO
local_enable=YES

```
write_enable=NO
anon_upload_enable=NO
anon_mkdir_write_enable=NO
anon_other_write_enable=NO
chroot_local_user=YES
guest_enable=YES
guest_username=virtual
listen=YES
listen_port=21
pasv_min_port=30000
pasv_max_port=3099
user_config_dir=/etc/vsftpd_user_conf
```

8) 建立虚拟帐户权限配置目录 mkdir /etc/vsftpd_user_conf

9) 建立账号 admin 管理员，和 guest

```
vim /etc/vsftpd_user_conf/admin              管理员账号
anon_world_readable_only=NO
write_enable=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES
```

```
vim /etc/vsftpd_user_conf/guest              普通权限账号
anon_world_readable_only=NO
```

10) 启动 vsftpd /etc/init.d/vsftpd start

11) 只准许 192.168.1.0/24 访问 ftp 服务器。

```
vim    /etc/vsftpd/iptables.sh
#!/bin/bash
ipt="/sbin/iptables"
$ipt -t filter -A INPUT -p tcp -s 192.168.1.0/24 --dport 21 -j ACCEPT
$ipt -t filter -A INPUT -p tcp --dport 21 -j DROP
```

12) 加入 rc.local ，开机并执行

```
vim    /etc/rc.local
```

/bin/sh /etc/vsftpd/iptables.sh
/etc/init.d/vsftpd start

第六章　WEB

6.1 apache 介绍

Apache 是世界使用排名第一的 Web 服务器。它可以运行在几乎所有广泛使用的计算机平台

上，官方网站 http://apache.org/。

6.2 Apache+php+mysql 编译与安装

yum 形式安装
[wds@localhost ~]# yum install httpd
源码编译

[wds@localhost ~]# tar -xvzf httpd-2.2.10.tar.gz
相关参数
[wds@localhost ~]# ./configure --prefix=/data1/apache2 / 安装地址
--enable-mods-shared=most /
--enable-ssl=shared --with-ssl=/user/local/ 支持 ssl
以下安装 subversion 需要装
--enable-so / 共享后安装的模块
--enable-maintainer-mode /
--enable-dav/ 安装 subversion 必须装
[wds@localhost ~]# make
[wds@localhost ~]# make install

Mysql 安装
[wds@localhost ~]# tar -xvzf mysql-5.1.50.tar.gz
[wds@localhost ~]# cd mysql-5.1.50
[wds@localhost ~]#./configure --prefix=/usr/local/mysql/ 安装路径
--without-debug/ 关闭 debug 模式
[wds@localhost ~]#.make
[wds@localhost ~]# make install

Openssl 安装
[wds@localhost ~]# tar -zxf openssl-0.9.8d.tar.gz
[wds@localhost ~]# cd openssl-0.9.8d
[wds@localhost ~]# ./config
[wds@localhost ~]# make && make test
[wds@localhost ~]# make install

Libiconv 安装
[wds@localhost ~]# ./configure && make && make check
[wds@localhost ~]# make install

[wds@localhost ~]# cd freetype-2.1.10
[wds@localhost ~]# ./configure && make
[wds@localhost ~]# make install

libpng-1.2.8-config
[wds@localhost ~]# cp -ip scripts/makefile.linux ./makefile
[wds@localhost ~]# ./configure && make
[wds@localhost ~]# make install

zlib-1.2.3
[wds@localhost ~]# ./configure
[wds@localhost ~]# make test
[wds@localhost ~]# make install

Gd 安装
[wds@localhost ~]# ./configure --with-png=../libpng-1.2.8-config --with-freetype=../freetype-2.1.10 --with-libiconv-prefix=../libiconv-1.9

Php 安装
[wds@localhost ~]# tar -xvzf php-5.2.6.tar.bz2
[wds@localhost ~]# cd php-5.2.6
[wds@localhost ~]# ./configure --with-apxs2=/data1/apache2/bin/apxs 指定 apache 的 apxs
[wds@localhost ~]# make
[wds@localhost ~]# make test
[wds@localhost ~]# make install

Mysql 扩展安装
[wds@localhost ~]# cd ./php-5.2.6/ext/mysql
[wds@localhost ~]# phpize
[wds@localhost ~]# ./confgure ---prefix=/usr/local/mysql
[wds@localhost ~]# make
[wds@localhost ~]# make install

apachetop-0.12.6
[wds@localhost ~]# ./configure
[wds@localhost ~]# make
[wds@localhost ~]# make install

6.3 httpd 的套件结构

/bin 二进制命令目录
/conf httpd 配置文件
/htdocs web 发布目录

/logs　　程序日志目录
/lib　　库文件
/manual　　文档
/modules　　扩展模块
/include　　头文件

6.4 httpd.conf 文件介绍

ServerRoot "/etc/httpd"
#定义了配置文件，错误文件的目录地址

PidFile run/httpd.pid
#记录 Apache 进程号

Timeout 120
#超时是控制客户端和与连接超时的秒数

KeepAlive Off
#实现 HTTP1.1 版本的连接功能，就是说什么浏览器都能浏览

MaxKeepAliveRequests 100
#可以连接的最大客户端

KeepAliveTimeout 15
#一次连接的多次请求，比如说一次连接后断开后，在连接就要超过这个时间才可以进行下次连接

<IfModule prefork.c>
StartServers　　8　　#用来设置 httpd 启动子进程的数量，
MinSpareServers　　5　　#应该设置 StartServers 和 MaxSpareServers 之间的数
MaxSpareServers　　20
ServerLimit　　256
MaxClients　　256　　# 最大的客户端连接数
MaxRequestsPerChild　　4000　　# 一个子进程为多次连接为服务
</IfModule>

#Listen 12.34.56.78:80

Listen 80
#定义默认访问的端口
LoadModule suexec_module modules/mod_suexec.so
LoadModule disk_cache_module modules/mod_disk_cache.so
LoadModule file_cache_module modules/mod_file_cache.so
LoadModule mem_cache_module modules/mod_mem_cache.so
LoadModule cgi_module modules/mod_cgi.so

#以上为加载的所有模块
#
Load config files from the config directory "/etc/httpd/conf.d".

Include conf.d/*.conf
#载入 conf.d 下的文件

User apache
Group apache
#apache 使用的用户组和名

ServerAdmin root@localhost
#如果 Apache 发生了什么问题，ServerAdmin 会给 root 发邮件

#ServerName new.host.name:80
#服务器 dns 名

DocumentRoot "/data1/apache/sso/htdocs"
#默认的访问首页地址

DirectoryIndex index.html index.html.var
执行文件的顺序

#ErrorLog logs/error_log
#系统错误日志存放位置

ServerSignature On
关闭 apache 404、500 错误提示
AddDefaultCharset UTF-8
#修改网页字体文件的位置，如果你的网页出现乱码就在这里把 UTF-8 该成 "AddDefaultCharset GB2312"

6.5 虚拟主机

6.5.1 基于 ip 虚拟主机

```
<VirtualHost 192.168.0.9>
        serverName 192.168.0.9:80
         DocumentRoot /data1/apache/htdoc/www1
        ServerName localhost
        ServerAdmin 7717060@sina.com.
</VirtualHost>
#-----------------------------------------------------------------
<VirtualHost 192.168.0.10>
        ServerName 192.168.0.10:80
        DocumentROOT /data1/apache/htdoc/www2
        ServerName localhost
        ServerName 7717060@sina.com
</VirtualHost>
#-----------------------------------------------------------------
<VirtualHost 192.168.0.11:8080>
        ServerName 192.168.0.11:8080
        DocumentRoot /data1/apache/htdoc//www3
        ServerName localhost
        ServerName 7717060@sina.com
</VirtualHost>
```

6.5.2 基于域名的虚拟主机

```
NameVirtualHost 74.82.166.60:80

<VirtualHost 74.82.166.60:80>
            DocumentRoot "/data1/apache2/htdocs/test/"
            DirectoryIndex "/index.php"
            ServerName test.365online.me

            LogFormat "i %h %l %u %t \"%r\" %>s %b \"%{Referer}i\""
            TransferLog "|/data1/apache2/bin/rotatelogs
/data1/apache2/logs/%Y%m/test.365online.me.80-access_log.%Y%m%d 86400 480"
            ErrorLog " |/data1/apache2/bin/rotatelogs
/data1/apache2/logs/%Y%m/test.365online.me.80-error_log.%Y%m%d 86400 480"
```

</VirtualHost>

6.6 如何给 apache 安装模块

开发网站和网页时,客户端为加快速度经常一些网页内容来加快访问速度，这样在开发实时性网站时就遇见了一些困难，所以程序员要在 http 的 header 头中加入（Header set Cache-Control "no-cache, no-store, private, must-revalidate"）信息让留言器不缓存一数据，其实这一点 apache 也可以做到，加入 herader 模块就可以了，在网上下载该模块 mod_headers.c

[wds@localhost]./apache/bin/apxs -i -a -c mod_headers.c
[wds@localhost]/apache/bin/httpd -l
就可看见已经加入的模块。同时将以下两句句话加入到 httpd.conf 中

LoadModule headers_module libexec/mod_headers.so

<FilesMatch "\.php$">
Header set Cache-Control "no-cache, no-store, private, must-revalidate"
</FilesMatch>

6.7 apache+Php+mysql 安装

6.8 apache 配置 subversion

subversion 下载地址：http://subversion.tigris.org/

1.安装 subversion

[wds@localhost ~]# tar xvjf subversion-1.6.3.tar
[wds@localhost ~]# ./configure --prefix=/data1/subversion/ --with-apxs=/data1/apache2/bin/apxs/
  --with-apr-util=/data1/apache2/bin/apu-1-config/
  --with-apr=/data1/apache2/bin/apr-1-config

在以上编译过程中可能会出现 configure: error: Subversion requires SQLite
可以 http://www.sqlite.org/sqlite-amalgamation-3.6.13.tar.gz 可以下载到最新的 Sqlite 版本 sqlite-amalgamation-3.6.13.tar.gz 安装 Sqlite3.6.13 tar zxvf sqlite-amalgamation-3.6.13.tar.gz 进入文件夹 sqlite-amalgamation 找到 sqlite3.c 将其复制到
subversion-1.6.2/sqlite-amalgamation/sqlite3.c

[wds@localhost ~]# make && make install

3.环境配置

mkdir -p /data1/subversion/
svnadmin create /data1/subversion/bsso

4.配置 apache

```
<Location /svn>
    DAV svn
    SVNParentPath /data1/subversion/bsso
    AuthType Basic
    AuthName "Subversion repository"
    AuthUserFile /data1/subversion/passwd
    Require valid-user
</Location>
```

/data1/apache2/bin/htpasswd -c /data1/subversion/passwd sina_sso

第七章mysql 服务器

7.1 mysql 介绍

MySQL 是一个小型关系型数据库管理系统，开发者为瑞典 MySQL AB 公司。在 2008 年 1 月 16 号被 Sun 公司收购。目前 MySQL 被广泛地应用在 Internet 上的中小型网站中。由于其体积小、速度快、总体拥有成 本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了 MySQL 作为网站数据库。MySQL 的官方网站的网址 是：www.mysql.com .

7.2 mysql 安装

[wds@localhost ~]# yum install mysql-server

[wds@localhost ~]# tar -xvzf mysql-5.1.50.tar.gz
[wds@localhost ~]# cd mysql-5.1.50
[wds@localhost ~]# ./configure --prefix=/usr/local/mysql/
                                --without-debug/

[wds@localhost ~]# make
[wds@localhost ~]# make install
[wds@localhost ~]# cd /usr/local/mysql/bin/mysql
[wds@localhost ~]# useradd mysql
[wds@localhost ~]# ./mysql_install_db
[wds@localhost ~]# chown -R mysql:mysql /usr/local/mysql
[wds@localhost ~]# ./ usr/local/mysql/bin/mysqld_safe &
[wds@localhost ~]# ./usr/local/mysql/bin/mysql

7.3 常用 mysql 命令

1、说明：创建数据库
Create DATABASE database-name
2、说明：删除数据库
drop database dbname
3、说明：备份 sql server
--- 创建 备份数据的 device
USE master
EXEC sp_addumpdevice 'disk', 'testBack', 'c:\mssql7backup\MyNwind_1.dat'
--- 开始 备份
BACKUP DATABASE pubs TO testBack
4、说明：创建新表
create table tabname(col1 type1 [not null] [primary key],col2 type2 [not null],..)
根据已有的表创建新表：
A：create table tab_new like tab_old (使用旧表创建新表)
B：create table tab_new as select col1,col2… from tab_old definition only
5、说明：删除新表
drop table tabname
6、说明：增加一个列
Alter table tabname add column col type
注：列增加后将不能删除。DB2 中列加上后数据类型也不能改变，唯一能改变的是增加 varchar 类型的长度。

7、说明：添加主键： Alter table tabname add primary key(col)
说明：删除主键： Alter table tabname drop primary key(col)
8、说明：创建索引：create [unique] index idxname on tabname(col….)
删除索引：drop index idxname
注：索引是不可更改的，想更改必须删除重新建。
9、说明：创建视图：create view viewname as select statement
删除视图：drop view viewname
10、说明：几个简单的基本的 sql 语句
选择：select * from table1 where 范围
插入：insert into table1(field1,field2) values(value1,value2)
删除：delete from table1 where 范围
更新：update table1 set field1=value1 where 范围
查找：select * from table1 where field1 like '%value1%' ---like 的语法很精妙，查资料!
排序：select * from table1 order by field1,field2 [desc]
总数：select count * as totalcount from table1
求和：select sum(field1) as sumvalue from table1
平均：select avg(field1) as avgvalue from table1
最大：select max(field1) as maxvalue from table1
最小：select min(field1) as minvalue from table1

7.4 恢复与备份数据库
[wds@localhost ~]# mysqldump -u 用户明 -p 密码 库名 > sql.bak 备份数据库
[wds@localhost ~]# mysql -u 用户明 -p 密码 库名 < sql.bak 恢复数据库

第八章 DNS 服务器

8.1 dns 常用网站

http://www.icann.org/ 顶级域名的管理组织
http://www.isc.org/ BIND 软件官方网站

http://www.isc.org/ml-archives/ BIND 的邮件列表，有很多问题可以在列表中找到你要的答案
http://www.internic.net/ Internet 域名注册
http://www.internic.net/regist.html 世界各地区的注册代理商数据
http://www.internic.net/whois.html 域名信息 whois 数据查询，国家后缀除外
http://www.allwhois.com/ 注册代理商、whois 数据查询
http://www.cnnic.cn/ .cn 后缀的域名服务，需要注册 .cn 后缀可以从这里开始
http://www.apnic.net/index.html 亚太地区网络 IP 注册查询服务
http://www.ripe.net/ 欧洲、中东、非洲等地区网络 IP 注册查询服务
http://www.arin.net/ 南美地区网络 IP 注册查询服务
http://www.dnsstuff.com/ dns online tools 可以查询很多有用数据，特别是反垃圾邮件的信息
http://www.dnsreport.com/ 可以对你的域名进行全面的测试，并提供完成的报告哦

8.2Dns 简介

Dns 全称（Domain Name System）是 internet 和 intranet 中重要的网络服务，它是一种组织成域层次的计算机和网络服务命名系统，人们可以通过简单好记的域名来代替 ip 访问网络

8.3 编译 dns

[wds@localhost ~]# yum -y install bind*

[wds@localhost ~]# yum -y install caching-nameserver

8.4 相关文件

1) /etc/hosts

Hosts 在 linux 中解析域名时先通过 hosts 解析，如果域名不存在在去互联网解析，好处减少带宽。而且使用 hosts 文件还可以减少对 DNS 服务器的访问来加快访问速度并减少带宽消耗。

2) /etc/host.conf

如果想调整 linux 的 dns 解析顺序可以修改/etc/hosts.conf.

3) /etc/resolv.conf

通过修改 resolv.conf，可以让 linux 通过哪个 dns 去解析域名(google 出了自己的 dns，地址为 8.8.8.8可以试验一下效果不错)

4) named.conf 主配置文件

开始配置文件：
加入以下黑体字的内容

[wds@localhost ~]# vi /etc/named.conf　　添加以下内容

options {

listen-on port 53 { 127.0.0.1; };　　#监听主机的 ip

fetch-glue no;

recursion no;

directory "/var/named";

dump-file "/var/named/data/cache_dump.db";

statistics-file "/var/named/data/named_stats.txt";

```
};
controls {
        inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
zone "." IN {
        type hint;
        file "named.ca";
};
zone "blog.365online.me" IN {
        type master;
        file "blog.365online.me";
        allow-update { none; };
};
zone "0.168.192.in-addr.arpa"    IN {
        type master;
        file "0.168.192.in-addr.arpa";
        allow-update { none; };
};
include "/etc/rndc.key";
```

5) 正向域

[wds@localhost ~]# vim /var/named/chroot/var/named/blog.365online.me

```
1) $TTL        86400
2) @                    IN SOA    blog.365online.me.    root.blog.365online.me .(
3)                                                    42                    ; serial
4)                                                    3H                    ; refresh
5)                                                    15M                   ; retry
6)                                                    1W                    ; expiry
7)                                                    1D )                  ; minimum

8)                      IN NS                ns1.blog.365online.
9)                      IN MX                10      blog.365online.
10) www                  IN        A          192.168.0.8
```

第 1 行，这行内容给出了该域名(blog.365online.me)各种记录的默认 TTL 值，这里为 1 天。即如果该域名的记录没有特别定义 TTL，则默认 TTL 为有效值。
第 2 行，指定 SOA 记录，因为主配置文件中定义名为 blog.365online.me,所以这里@代表 blog.365online.me.
第 3 行，从这行开始到第 7 行为该域名的 SOA 记录部分，这里的@代表域名本身。
第 3 行，Serial 部分，这部分用来标记 ZONE 文件更新，如果发生更新则 Serial 要单增，否则 MASTER 不会通知 SLAVE 进行更新。
第 4 行，Refresh 部分，这个标记 SLAVE 服务器多长时间主动(忽略 MASTER 的更新通知)向 MASTER 复核 Serial 是否有变，如有变则更新之。
第 5 行，Retry 部分，如 Refresh 过程不能完成，重试的时间间隔。
第 6 行，Expire 部分，如 SLAVE 无法与 MASTER 取得联系，SLAVE 继续提供 DNS 服务的时间，这里为 2W(两周时间)。Expire 时间到期后 SLAVE 仍然无法联系 MASTER 则停止工作，拒绝继续提供服务。Expire 的实际意义在于它决定了 MASTER 服务器的最长下线时间(如 MASTER 迁移，DOWN 机等)。
第 7 行，Minimum 部分，这个部分定义了 DNS 对否定回答(NXDOMAIN 即访问的记录在权威 DNS 上不存在)的缓存时间。
第 8~13 行 详细见名词解释

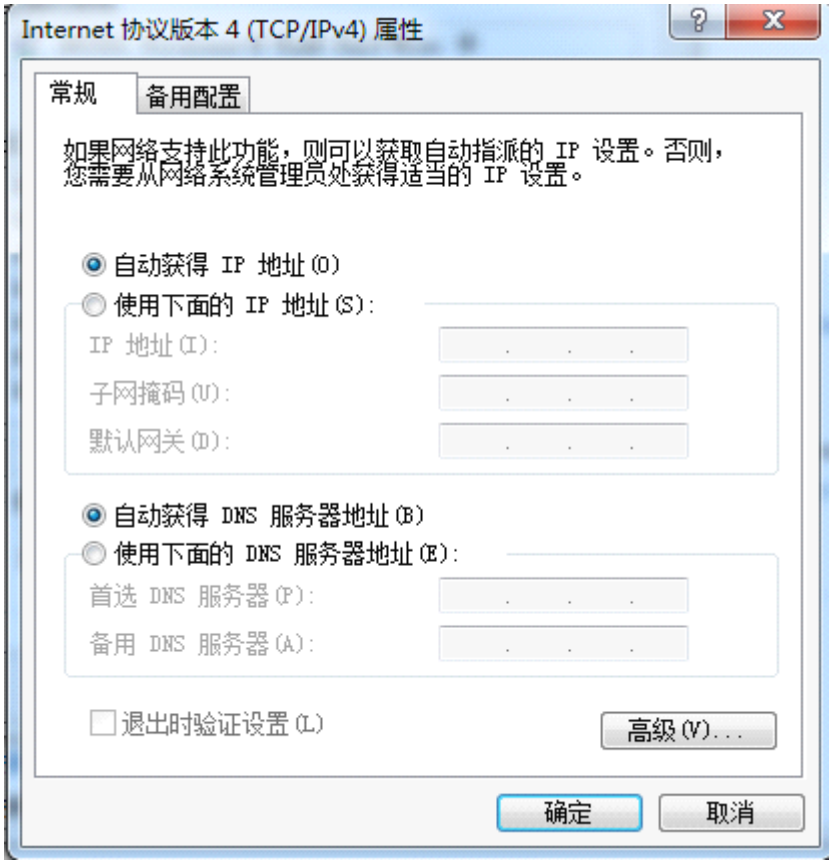
名词解释：
A 记录：记录域名到 IP 之间的关联。
MX 记录：定义了发往 XXX@ABC.COM 邮箱的邮件服务器地址。

6) 加入反向域

```
[wds@localhost ~]# vim    /var/named/chroot/var/named/0.168.192.in-addr.arpa
$TTL       86400
@          IN          SOA         blog.365online.me. hostmaster.365online.me.    (
                                                  1997022700 ; Serial
                                                  28800          ; Refresh
                                                  14400          ; Retry
                                                  3600000        ; Expire
                                                  86400 )        ; Minimum

8.0.168.192.in-addr.arpa.              IN          PTR         www.blog.365online.me.
```

7) windows 下修改 dns 配置,如图.



第十章 Squid

1.Squid 是什么?
2.Squid 能做什么?
3.Squid 如何工作的?
4.Squid 的工作原理是什么?
5.Squid 实例讲解

11.1 squid 是什么

Squid 是一个高性能的代理缓存服务器，Squid 支持 FTP、gopher 和 HTTP 协议。

和一般的代理缓存软件不同，Squid 用一个单独的、非模块化的、I/O 驱动的进程来处理所有的客户端请求。

Squid 将数据元缓存在内存中，同时也缓存 DNS 查询的结果，除此之外，它还支持非模块化的 DNS 查询，对失败的请求进行消极缓存。Squid 支持 SSL，支持访问控制。由于使用了 ICP（轻量 Internet 缓存协议），Squid 能够实现层叠的代理阵列，从而最大限度地节约带宽。

Squid 由一个主要的服务程序 squid,一个 DNS 查询程序 dnsserver，几个重写请求和执行认证的程序，以及几个管理工具组成。当 Squid 启动以后，它可以派生出预先指定数目的 dnsserver 进程，而每一个 dnsserver 进程都可以执行单独的 DNS 查询，这样一来就大大减少了服务器等待 DNS 查询的时间。

11.2 Squid 能做什么

11.2.1 代理服务器

在一般情况下，我们使用网络浏览器直接去连接其他 Internet 站点取得网络信息时，是直接联系到目的站点服务器，然后由目的站点服务器把信息传送回来。代理服务器是介于客户端和 Web 服务器之间的另一台服务器，有了它之后，浏览器不是直接到 Web 服务器去取回网页而是向代理服务器发出请求，信号会先送到代理服务器，由代理服务器来取回浏览器所需要的信息并传送给你的浏览器。

代理服务器的作用在网址框中输入您要访问的网站地址，点击代理浏览便会打开新的窗口链接代理服务器，等待几秒即可，如果此时出现无法链接服务器等错误，请在上面尝试选择其它的服务器，因为代理服务器对资源的消耗比较大，并且存在时效性，因此有时候无法打开，必须多次尝试代理服务器。每天自动更新最新可用服务器。大部分代理服务器都具有缓冲的功能，就好像一个大的 Cache，它有很大的存储空间，它不断将新取得数据储存到它本机的存储器上，如果浏览器所请求的数据在它本机的存储器上已经存在而且是最新的，那么它就不重新从 Web 服务器取数据，而直接将存储器上的数据传送给用户的浏览器，这样就能显著提高浏览速度和效率。

11.2.2 透明代理

透明代理的意思是说用户在一企业内不用在 IE 中设置代理服务器地址,通过默认网关的形式就可以上网，透明代理结合了 squid+ iptables ,让用户的体验更加完美些。

11.2.3 反向代理

反向代理（Reverse Proxy）方式是指以代理服务器来接受 internet 上的连接请求，然后将请求转发给内部网络上的服务器，并将从服务器上得到的结果返回给 internet 上

请求连接的客户端，此时代理服务器对外就表现为一个服务器。

通常的代理服务器，只用于代理内部网络对 Internet 的连接请求，客户机必须指定代理服务器,并将本来要直接发送到 Web 服务器上的 http 请求发送到代理服务器中。由于外部网络上的主机并不会配置并使用这个代理服务器，普通代理服务器也被设计为在 Internet 上搜寻多个不确定的服务器,而不是针对 Internet 上多个客户机的请求访问某一个固定的服务器，因此普通的 Web 代理服务器不支持外部对内部网络的访问请求。当一个代理服务器能够代理外部网络上的主机，访问内部网络时，这种代理服务的方式称为反向代理服务。此时代理服务器对外就表现为一个 Web 服务器，外部网络就可以简单把它当作一个标准的 Web 服务器而不需要特定的配置。不同之处在于，这个服务器没有保存任何网页的真实数据，所有的静态网页或者 CGI 程序，都保存在内部的 Web 服务器上。因此对反向代理服务器的攻击并不会使得网页信息遭到破坏，这样就增强了 Web 服务器的安全性。

第十一章 内核+iptables 升级

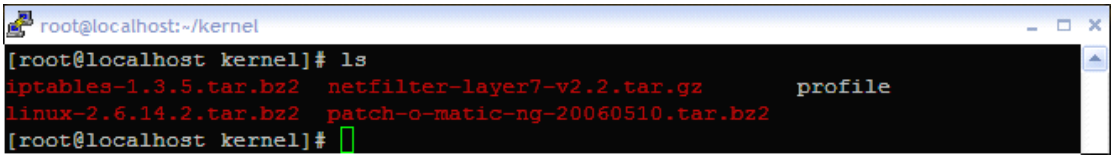
现在升级内核的方法有很多比如原码升级，rpm 升级，yum 升级等,我觉的还是原码升级比较专业，可以自己定制自己的内核。

11.1 编辑环境

虚拟机 VMware Workstation 6.0.0 build-
Red hat 4
Kernel version 2.6.-9-42
Iptables v1.2.11

11.2.下载原码包

官方网站（可以得到最新的补丁和内核，还有安装说明）：
http://www.kernel.org/pub/linux/kernel/v2.6/
http://ftp.netfilter.org/pub/iptables/
http://ftp.netfilter.org/pub/patch-o-matic-ng/snapshot/
http://sourceforge.net/projects/l7-filter/
下载后解压缩



11.3 基本安装顺序

0）配置环境变量
1）生成.config 文件
2）给内核打补丁
3）给防火墙打补丁
4）给防火墙增加新特性
5）make menuconfig，把新的选项选上（即第二步“打进去”的选项），保存退出

6）make
7）make modules_install
8）make install
9）编辑/boot/grub/grub.conf，把默认启动改为 0（即选择新的内核启动）
10）reboot
11）升级 iptables

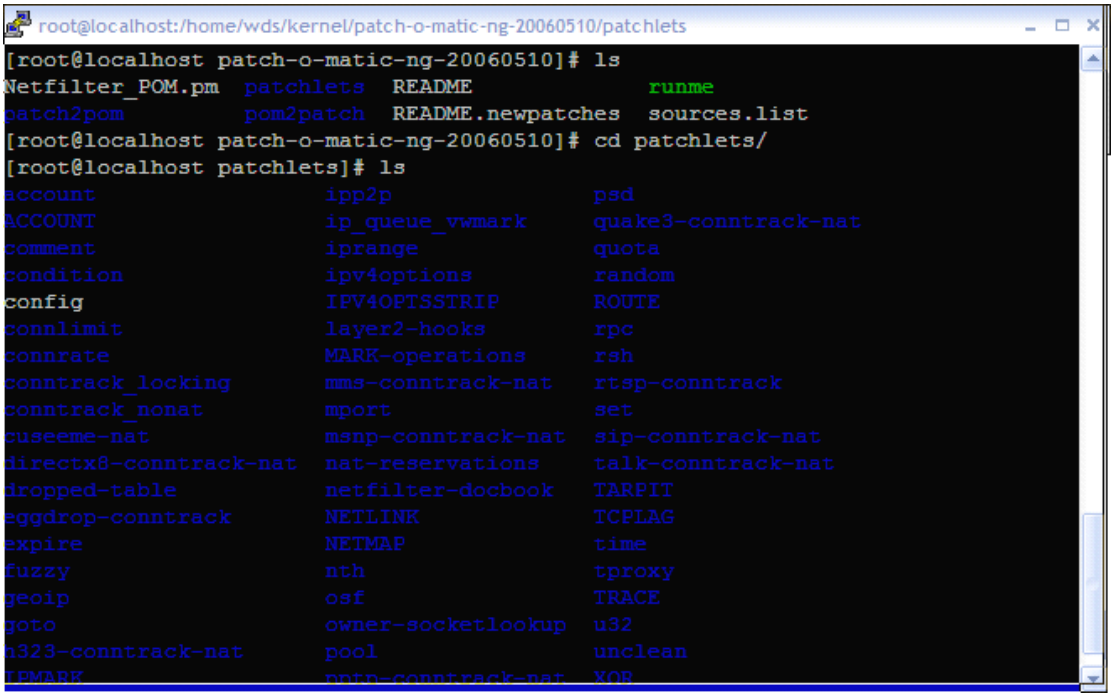
0) vi profile
export KERNEL_DIR=**/home/wds/kernel**/linux-2.6.14.2
export IPTABLES_DIR=**/home/wds/kernel/**iptables-1.3.5
export PATCH_O_MATIC_NG=/**home/wds/kernel**/patch-o-matic-ng-20060510
黑体字的地方是你放置原码的地方
[wds@localhost kernel]# source profile 倒入环境变量
[wds@localhost kernel]# set | more 查看是否倒入

1）生成.config 文件
这一步也不是必需的，只是为了方便以后的步骤，因为.config 文件的作用是纪录"哪些选项选了，哪些选项没有选"的，仅此而已，如果有兴趣你可以研究一下 makefile，看看.config 有什么用最简单生成.config 的方法就是执行 make menuconfig，然后什么都不做，保存退出也可以 cp /boot/config-2.6.xxx ./.config

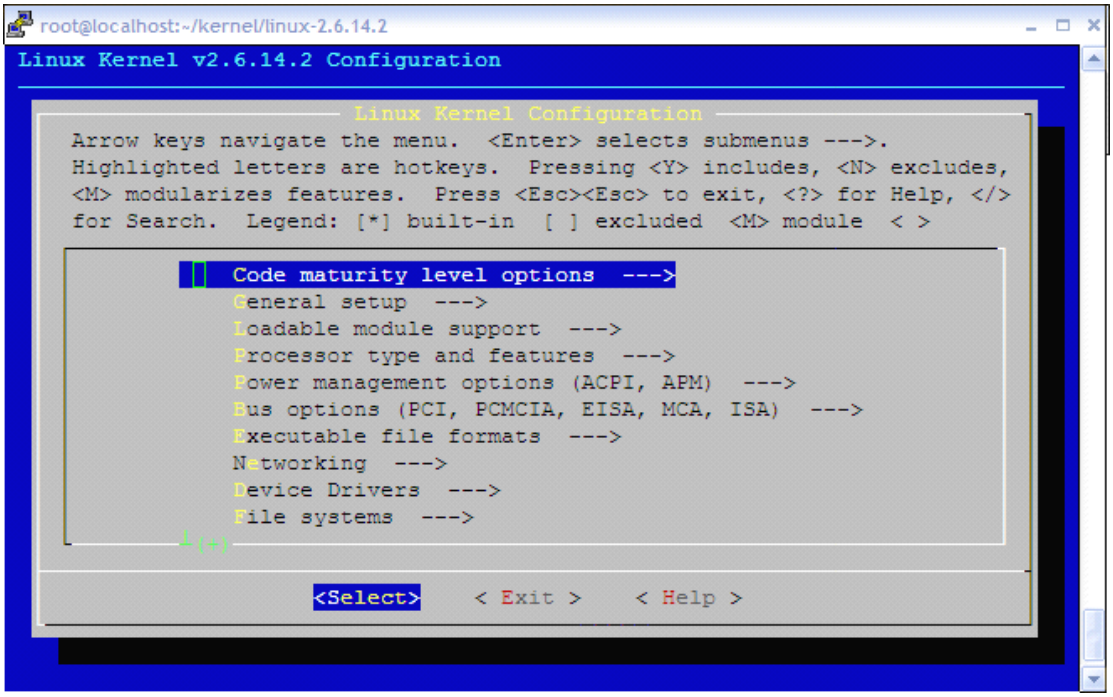
2) 给内核打补丁
进入内核文件夹(linux-2.6.14.2)
patch –p1 <../netfilter-layer7-v2.2/kernel-2.6.13-2.6.16-layer7-2.2.patch

3) 给防火墙打补丁
进入防火墙目录(iptables-1.3.5):
patch -p1 < ../netfilter-layer7-v2.2/iptables-layer7-2.2.patch

4) 增加新特性
进入(patch-o-matic-ng-20060510)
里面有个 patchlets 文件夹为新特性文件夹可以进去看一下都有什么新特性
退回到上一级目录进行新特性的安装，我主要使用 ipp2p(防 p2p 软件)
Time(按时间过滤) connlimit(控制并发) random(随机丢包)
./runme time
./runme ipp2p
./runme connlimit
./runme random



5) make menuconfig



进入内核目录输入 make menuconfig 命令会出现上边的图片,具体按下边配置

1. 有 SCSI 卡(如 VMware 中的 BusLogic BT946C) 的要把它编译成模块(M) 不然

会在 make install 出错 。 （其它关于 SCSI 的选项为内核内建，Buslogic 可以选择为模块，只是最后需要制作 initrd 模块）

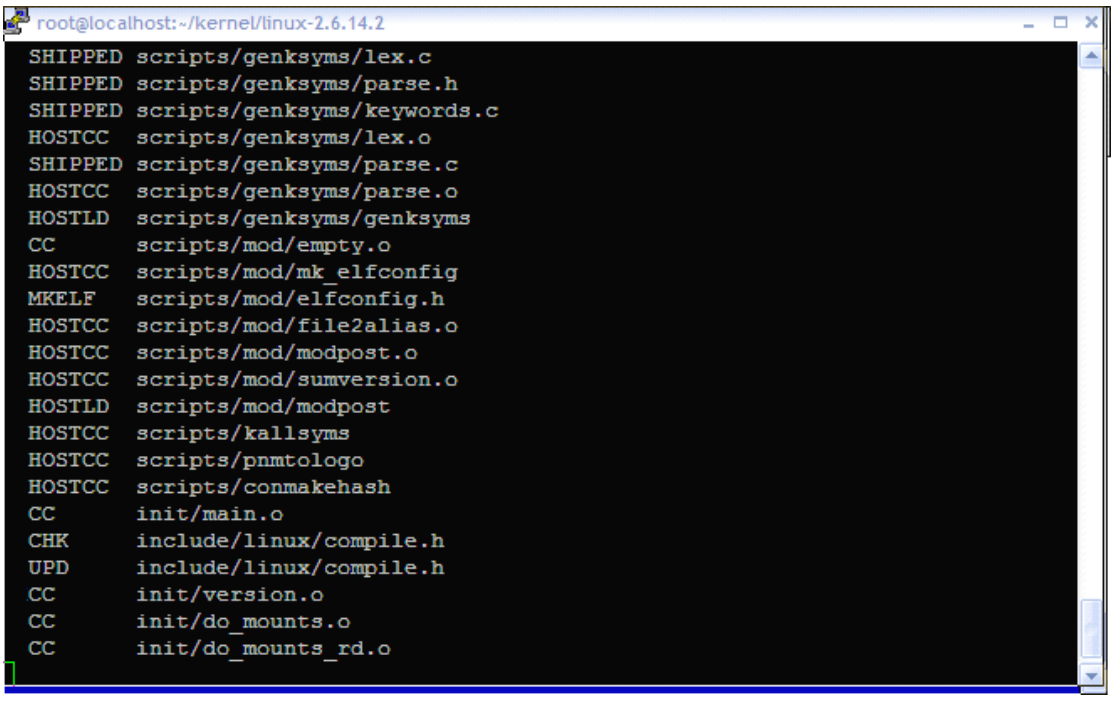
Device Drivers --->SCSI device support --><*> SCSI disk support

Device Drivers --->SCSI device support --->SCSI low-level drivers ---> <*> BusLogic SCSI support

2. Linux Kernel Configuration ---> Networking ---> Networking options ---> Network packet filtering (replaces ipchains) ---> IP: Netfilter Configuration ---> 你会发现下面多了几个后面带"（NEW）"字眼的，这就是我们刚才"打进去"的补丁.
3. 其他选项请参照网上帮助.

6) make

保存后退出.并执行 make 命令如果你的虚拟机内存大的话可以在 make 后加参数 –j 后加数字表示增加多少的线程.



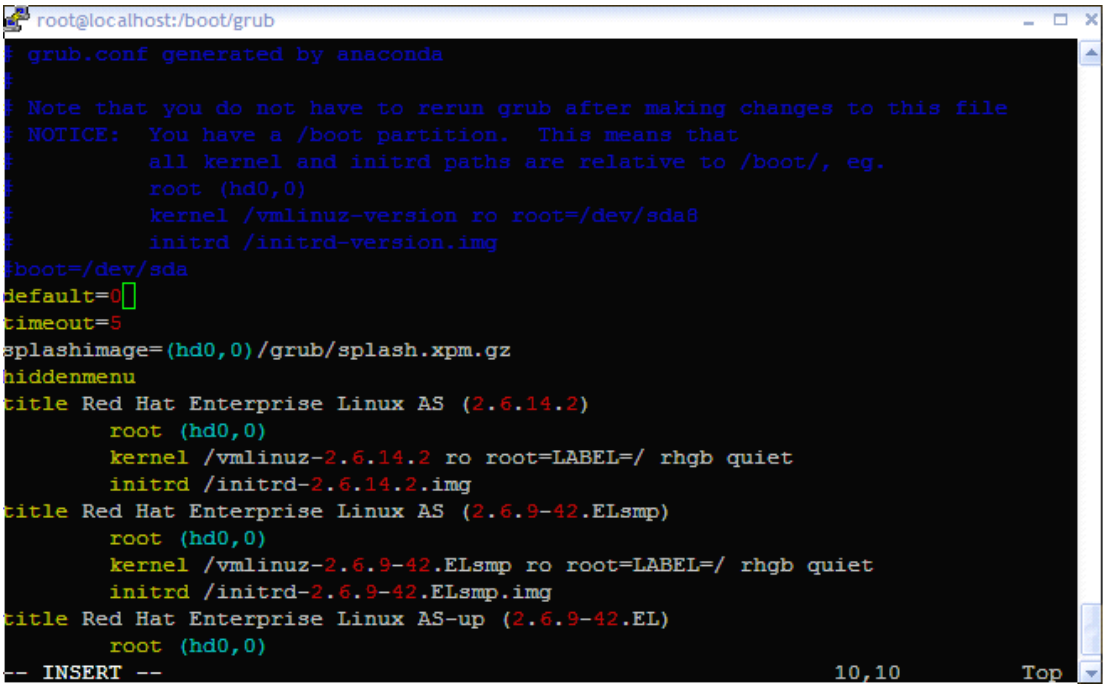
7) make modules_install

编译模块.....

8) make install

安装.....

9）编辑/boot/grub/grub.conf，把默认启动改为 0（即选择新的内核启动）



10) reboot 从新启动计算机