

第6章 建立Internet 站点

6.1 E-mail服务器的建立

E-mail 以其快捷方便的特性成为人们使用最多的工具，安装设置 Mail服务器软件是一个必不可少的部分。本章详细介绍电子邮件服务器软件的安装设置及其使用中的一些技巧。常见的电子邮件服务器有三种：Sendmail服务器、邮局协议 (POP) 邮件服务器和 Unix-to-Unix Copy (UUCP) 服务器。

Sendmail服务器处理Internet 中来回传输的SMTP邮件。SMTP为简单邮件传输协议，它是从一个站点到另一个站点传输邮件的标准协议。

POP服务器如同邮局一样地保管信件。用户可以直接和 POP服务器连接，它将负责将邮件传送给他们。

UUCP服务器 的性质介于SMTP及POP之间。

6.1.1 sendmail 服务器的设置

Sendmail对一般的系统管理者而言，往往是个难点，因为系统的使用者对 E-mail的需求与依赖程度高，不能有分毫差错。

Sendmail 8.9.3.1是目前Sendmail 8.9.x系列的最高版本。系统安全方面，是目前评价最好的，目前已知的BUG都已经改好了。因为本软件是一 public domain，可以在网路上各 ftp server上找到其source code。由于该软件的发展者把一些必要的设定步骤自动化了，所以，其实安装这个软件并不是非常的困难。

笔者把Sendmail的安装分成几个步骤：

1) 获取此软件：(通过ftp相关站点或直接从安装光盘上获取)，至少要有

```
sendmail.8.9.3.1.base.tar.gz  
sendmail.8.9.3.1.cf.tar.gz  
sendmail.8.9.3.1.xdoc.tar.gz
```

这三个文件，其中，第一个文件是此软件的程序；第二个文件是此软件的设定，此二者是必要的文档；第三个是文本文件，如果不想看，可以不要。得到以上文件以后，解开文件。建议管理者读一读此软件的相关文件。

2) 全解压开后应该会在该目录下建立几个目录，其中程序是在“~/sendmail/src”的目录下，设定文件在“~/sendmail/cf”目录下。请1转换至“~/sendmail/src”的目录，用ls命令可以看到许多名 Makefile.xxx 的文件档案。请使用者找到一个适合自己系统的 makefile，如：假设是用IBM AIX的系统，那就使用Makefile.AIX；用HP的系统，那就用Makefile.HP-UX；如果是Sun系统，那就用 Makefile.SunOS.x.x，但如果是 SunOS 4.1.3，就得要用 Makefile.SunOS这个makefile文件，而如果是Solaris 2.3，则得要用 Makefile.SunOS.5.x。用vi或任何其他编辑器，调出刚挑出的makefile文件，做一点点必要的修改。所要修改的地方不多，

下面是可能要修改的地方：

- "CC ="如果原文件所用的 compiler并不是自己系统的，这才要改。
- 本系统是否有安装Berkeley 的New database？如果有，跳过本点，到第3点。如果没有（如：一般SunOS 都没有装），那么请修改这几行（下面以Makefile.SunOS为例，但其他系统也类似）：

a.在 DBMDEF= -DNDBM -DNEWDB -DNIS 一行，请改成：

```
DBMDEF= -DNDBM -DNIS(即去掉-DNEWDB)
```

b.把这两行REMARK起来：

```
INCDIRS=-I/usr/sww/include/db
--> # INCDIRS=-I/usr/sww/include/db
LIBDIRS=-L/usr/sww/lib
--> # LIBDIRS=-L/usr/sww/lib
```

c.修改LIBS= -ldb -ldbm -lresolv 这一行成：

```
LIBS= -ldbm -lresolv
```

(即去掉-db)

把nroff的“-mandoc”参数改成一般系统可以接受的“-man”(当然，这得依你的系统而定)。如果不改也并不会影响sendmail的安装，只是不能看到格式化后的sendmail相关文件罢了。

3) 输入make -f Makefile.xxx等待编译完毕。不过，笔者可以乘这个空文件把旧的 sendmail 相关文件案备份起来，以防万一有什么不测，还可以有所储备)。要备份的文件是：/usr/lib/sendmail 和 sendmail.cf(后者所在位置得依原系统而定，不过，一般都是在 /etc下或 /etc/mail下)。有一点要记住的是，原来的 /usr/lib/sendmail 是个suid root的程序，备份的请记住住执行chmod u-s xxxx，要不然，留着这样一个suid root 的程序，也是会让人有机可乘的！

4) 此一步骤也可以在 compile 程序时进行，就是要产生系统设定文件，即 sendmail.cf。

请切换至设定文件目录 ~ /sendmail/cf/cf 之下。找一个合适自己系统的 xxxx.mc文件。笔者一般是使用 tcpproto.mc 这一个不用uucp的文件。如果是用 tcpproto.mc 这个文件，笔者在 solaris 2.3试了，在文件中要加上一行：OSTYPE(solaris2)，如果不小心忘了加上OSTYPE()，会有“unknow mailer”的错误，然后信寄不出去。

选定了某一个合意的.mc文件，接下来可执行m4产生.cf文件。请用下面指令：

```
# m4 XXX.mc > YYY.cf
```

如果是sun 4.1.3，可能系统会抱怨m4的版本太旧了。请试用：

```
# /usr/5bin/m4 XXX.mc >YYY.cf
```

若还是不行，那就去找新版的m4回来compile。

产生出来的YYY.cf一般得要改某个地方才能正常地收信。请寻找“Cw”，原来产生的.cf文件，Cw应是：

```
Cwlocalhost
```

请在localhost之后加上你那一台机器所有可能用来当收信地址的 host name，如：

```
Cwlocalhost tarn.ustc.edu.cn man.ustc.edu.cn
```

如果这个步骤忘了做，将会有从外面机器寄信来却收不到，机器会抱怨：

```
"configure error ...."
```

5) 这个步骤是要把 compile 好的东西 install 到系统上去了。假设 compile 的结果没有任何

错误，产生.cf文件过程也就没有什么值得注意的信息。所产生的.cf文件也改好了。

切换至目录 `~/sendmail/src`，下指令：

```
# make -f Makefile.XXX install
```

系统会把compile出来的 `~/sendmail/src/sendmail`与man document拷贝到它们该有的目录。

切换至目录 `~/sendmail/cf/cf`，把产生出来的YYY.cf文件拷贝到本系统 `sendmail.cf` 应该在的地方，如：`/etc`或`/etc/mail`。

把旧sendmail daemon 的process kill 掉。用 `ps -ax |grep sendmail |grep -v grep` 找出旧daemon的process(或 `cat /etc/sendmail.pid` 或 `cat /etc/mail/sendmail.pid`)，然后用 `kill -9 processID` kill 掉它。

启动新daemon。例如下指令：

```
/usr/lib/sendmail -bd -q30m
```

当然，保守一点，看旧的daemon怎么跑，照旧的来下参数应该没有问题。

6) 这个步骤主要用来测试：

- a. local 的收、发信。
- b. remote的收、发信。
- c. 最好当然能试试不同的domain 或甚至net 的收发情况。

6.1.2 在Linux机器上设置pop3代理

在Linux服务器上,http/https/ftp代理的应用非常广泛，squid大家都很熟悉，但是有些特殊情况下，你可能需要让一台Linux机器也做为pop3代理服务器，代理客户端从互联网上接收邮件，并传送邮件给客户工作站。

也许这种服务对于专线连到ISP的用户更适合，在有的公司的内部网中，需要从外部世界收信，可能你不能直接与外部世界通信，但能很容易地访问到你的防火墙机器。这时，在防火墙机器上安装pop3代理就起到了一种不可替代的作用。

软硬件环境：Redhat6.0，TIS通用防火墙代理服务程序 `plug-gw` (下载)，实现方法：

1) 安装pop3代理非常简单，仅需要下载上面提到的一个很小的程序 `plug-gw`即可，下载后放入`/usr/local/etc`目录下(需要执行权限)。

2) 这里假设你需要代理互联网上两个邮件服务器，一个是 `tarn.ustc.edu.cn`，另一个是 `pop3.263.net`。

3) 在`/usr/local/etc`目录下创建存取表文件 `netperm-table`，像下面这样：

```
root@sh-proxy etc # more netperm-table
plug-gw: port 110 192.168.13.* -plug-to
public.sta.net.cn-port110
plug-gw: port 114 192.168.13.21 -plug-to pop3.163.net -port110
root@sh-proxy etc # ls -l
total 24
-rwxrwxr-x 1 root root 314 Sep 14 17:58 netperm-table
-r-xr-xr-x 1 root root 22489 Dec 31 1998 plug-gw
```

注意 第一行允许192.168.13.0网段上任何机器使用该代理服务器接收 `public.sta.net.cn`上的邮件。第二行仅允许一台机器 `192.168.13.21`使用该代理服务器接

收pop3.263.net上的邮件。

4) 修改/etc/services，加下面的一行：

```
pop3proxy 114/tcp
```

5) 在命令行下运行两句程序，也可加入到/etc/rc.d/rc.local的最后，使得每次启动时装载：

```
# /usr/local/etc/plug-gw -daemon pop3 pop3 &  
# /usr/local/etc/plug-gw -daemon pop3proxy pop3proxy &
```

注意 请确信/etc/services中存在一行pop3 110/tcp，若前面的名字不是pop3，而是其他的如pop-3等，那么或者更新为 pop3 110/tcp，或者运行 /usr/local/etc/plug-gw -daemon pop-3 pop-3 &

6) 若在Windows客户端用outlook express收发邮件，若从public.sta.net.cn收信，仅需改pop3服务器为你的防火墙即可，若从pop3.263.net收信，还得更改标准pop3端口号为114，而不是110。

6.2 FTP服务器的建立

目前很多企业单位正在构架自己的 Intranet，FTP服务器是其中的一个服务支持，有很多单位还配置了基于UNIX平台的计算机，为了在网上能够提供FTP服务支持，专门购置基于UNIX平台的FTP Server软件，其实在基于UNIX平台的计算机系统中，采取一定的技术方法，就可以建立自己的FTPServer，实现网上文件上下载服务，无须再购置专门的软件，何乐而不为之呢？下面就其建立过程阐述如下：

1) 确定FTP Server的共享目录。

为FTP Server建立一个FTP工作目录，在此设定为/home/ftp

2) 编写FTP Server初始设置的Shell程序(Setup)。

用编辑工具(如vi)编写一个Shell程序，用于对系统进行设置。

```
#!/bin/sh  
case $# in  
0) ftphome=`grep '^ftp:' /etc/passwd | cut -d: -f6`  
;;  
1) if [ "$1" = "start" ]; then  
ftphome=`grep '^ftp:' /etc/passwd | cut -d: -f6`  
else  
ftphome=$1  
fi  
;;  
*) echo "Usage: $0 [anon-ftp-root]"  
exit 1  
;;  
esac  
if [ -z "${ftphome}" ]; then  
echo "$0: ftphome must be non-null"  
exit 2  
fi  
if [ "${ftphome}" = "/" -o "${ftphome}" = "/usr" ]; then
```

```
echo "$0: ftphome must not be / or /usr"
exit 2
fi
if [ ! -d ${ftphome} ]; then
mkdir ${ftphome}
fi
if [ ! -d ${ftphome}/usr/bin ]; then
    if [ ! -d ${ftphome}/usr/bin ]; then
mkdir -p ${ftphome}/usr/bin
fi
cp /usr/bin/ls ${ftphome}/usr/bin
chmod 111 ${ftphome}/usr/bin/ls
chown root ${ftphome}/usr/bin
chmod 555 ${ftphome}/usr/bin
if [ -r ${ftphome}/bin ]; then
mv -f ${ftphome}/bin ${ftphome}/Obin
fi
ln -s usr/bin ${ftphome}
if [ ! -d ${ftphome}/usr/lib ]; then
mkdir -p ${ftphome}/usr/lib
fi
if [ ! -d ${ftphome}/etc ]; then
mkdir -p ${ftphome}/etc
fi
cp /usr/lib/ld.so /usr/lib/ld.so.1 ${ftphome}/usr/lib
for lib in libc libdl libintl libw libnsl libsocket \
nss_nis nss_nisplus nss_dns nss_files
do
cp /usr/lib/${lib}.so.1 ${ftphome}/usr/lib
rm -f ${ftphome}/usr/lib/${lib}.so
    ln -s ./${lib}.so.1 ${ftphome}/usr/lib/${lib}.so
done
cp /usr/lib/straddr.so.2 ${ftphome}/usr/lib
rm -f ${ftphome}/usr/lib/straddr.so
ln -s ./straddr.so.2 ${ftphome}/usr/lib/straddr.so
cp /etc/passwd /etc/group /etc/netconfig ${ftphome}/etc
chmod 555 ${ftphome}/usr/lib/*
chmod 444 ${ftphome}/etc/*
chown root ${ftphome}/usr/lib ${ftphome}/etc
chmod 555 ${ftphome}/usr/lib ${ftphome}/etc
if [ ! -d ${ftphome}/dev ]; then
mkdir -p ${ftphome}/dev
fi
prefix="/devices/pseudo/mm@0:"
for device in zero
do
line=`ls -l ${prefix}${device} | sed -e 's/,/,/'`
major=`echo $line | awk '{print $5}'`
minor=`echo $line | awk '{print $6}'`
rm -f ${ftphome}/dev/${device}
mknod ${ftphome}/dev/${device} c ${major} ${minor}
```

```
done
prefix="/devices/pseudo/clone@0:"
for device in tcp udp ticotsord
do
line=`ls -l ${prefix}${device} | sed -e 's/,//'\`
major=`echo $line | awk '{print $5}'`
minor=`echo $line | awk '{print $6}'`
rm -f ${ftphome}/dev/${device}
mknod ${ftphome}/dev/${device} c ${major} ${minor}
done
chmod 666 ${ftphome}/dev/*
chown root ${ftphome}/dev
chmod 555 ${ftphome}/dev
#建立一个匿名FTP服务目录,并设立可读可写属性
if [ ! -d ${ftphome}/pub ]; then
mkdir -p ${ftphome}/pub
fi
chown ftp ${ftphome}/pub
chmod 777 ${ftphome}/pub
```

3) 编辑/etc/passwd和/etc/shadow文件。

设置匿名FTP帐号,按下面方法分别对/etc/passwd和/etc/shadow两个文件中增加如下数据:

在/etc/passwd文件中增加

```
ftp:x:30000:30000:Anonymous FTP:/home/ftp:/nosuchshell
```

在/etc/shadow文件中增加

```
ftp:NP:6445:::~:~:
```

4) 运行setup程序。

在运行setup程序之前,将该文件设置成执行文件。

```
#chmod +x setup
```

运行setup程序 #setup /home/ftp

以上步骤就完成了 FTP服务器的建立,即可使用 FTP服务器,可以用如 explorer、netscape。

navigator等浏览器对FTP服务器进行访问,进行下载和上载文件操作,使用非常方便。

6.3 WWW服务器的建立

Internet 下最热的服务便是WWW服务,WWW服务在当今的信息社会是必不可少的一种方式。无论在哪个领域,动态的html页面都是一种宣传和获取信息的介质。为什么不运行你的WWW服务器呢?在这一节里,笔者以最普遍使用的阿帕奇(Apache)服务器为例,详细讲解阿帕奇的安装、编译,以及所有实现不同功能的指令。你将学会如何配置一个功能强大的Web服务器,并启动和测试这个服务器。

配置Web服务器

在此过程中确保你的服务器配置得适合你的系统需要,一旦完成这一过程,便没有什么可担心的了。假如决定提供不同的特性集或改变系统,则需要修改配置。

1. 一般的httpd配置文件的规则

一共需要建立三个配置文件，所有的文件都遵循以下规则：

- 在文件中与大小写有关的只是文件路径和文件名。
- 使用#号注释掉一行。
- 当使用directive 命令时，每行只能放置一个指令。
- 为了使指令及接下来的数据一目了然，可以留下一些空白行，Web服务器将忽略它们。

2. 建立服务器配置文件

在安装Linux的过程中，通过RPM包，你将很轻松地装载阿帕奇服务器软件。进入你的安装目录，服务器的配置文件需要我们根据具体的需求来设置。要建立的第一个配置文件就是为Web服务器所建立的，先使用阿帕奇的http.conf作为模板，对配置中遇到的参数做一定的说明(可根据自己的需求做改动，此配置仅供参考)。

```
# ServerType is either inetd, or standalone.
ServerType standalone
# If you are running from inetd, go to "ServerAdmin".
# Port: The port the standalone listens to.
Port 80
# HostnameLookups: Log the names of clients or just their IP numbers
#   e.g.   www.apache.org (on) or 204.62.129.132 (off)
HostnameLookups off
# NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
# when the value of (unsigned)Group is above 60000;
# don't use Group nobody on these systems!
User nobody
Group nobody
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.
ServerAdmin root@localhost
# ServerRoot: The directory the server's config, error, and log files
# are kept in.
ServerRoot /etc/httpd
# BindAddress: You can support virtual hosts with this option. This option
# is used to tell the server which IP address to listen to. It can either
# contain "*", an IP address, or a fully qualified Internet domain name.
# See also the VirtualHost directive.
BindAddress *
# ErrorLog: The location of the error log file. If this does not start
# with /, ServerRoot is prepended to it.
ErrorLog logs/error_log
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn
# The location of the access logfile (Common Logfile Format).
# If this does not start with /, ServerRoot is prepended to it.
CustomLog logs/access_log common
```

```
# PidFile: The file the server should log its pid to
PidFile /var/run/httpd.pid
ScoreBoardFile /var/run/httpd.scoreboard
#LockFile /var/lock/httpd.lock
# Apache needs to construct a self-referencing URL (a url that refers back
# to the server the response is coming from) it will use ServerName and
# Port to form a "canonical" name. With this setting off, Apache will
# use the hostname:port that the client supplied, when possible. This
# also affects SERVER_NAME and SERVER_PORT in CGIs.
UseCanonicalName on
# Timeout: The number of seconds before receives and sends time out
# Timeout: The number of seconds before receives and sends time out
Timeout 300
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
KeepAlive On
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We reccomend you leave this number high, for maximum performance.
MaxKeepAliveRequests 100
# KeepAliveTimeout: Number of seconds to wait for the next request
KeepAliveTimeout 15
# It does this by periodically checking how many servers are waiting
# for a request. If there are fewer than MinSpareServers, it creates
# a new spare. If there are more than MaxSpareServers, some of the
# spares die off. These values are probably OK for most sites ---
MinSpareServers 8
MaxSpareServers 20
# Number of servers to start --- should be a reasonable ballpark figure.
StartServers 10
MaxClients 150
# MaxRequestsPerChild: the number of requests each child process is
MaxRequestsPerChild 100
# Proxy Server directives. Uncomment the following line to
# enable the proxy server:
#ProxyRequests On
# To enable the cache as well, edit and uncomment the following lines:
#CacheRoot /var/cache/httpd
#CacheSize 5
#CacheGcInterval 4
```

建立你自己的文件 `httpd/conf/httpd.conf`,按照上面的内容及文件中的说明,修改你自己的配置,现在对其中的一些选项说明如下。

1) `#SeverType`可以选择 `inetd`或`standalone`

因为运行的是Internet下的站点,很容易选到 `inetd`,然而,要想要 `httpd`可以自我处理以占用较少的CPU资源,则可选择 `standalone`,因此我的 `httpd.conf`文件的这个选项行为:

```
SeverType standalone
```

2) `#Port`: The port the standalone listens to

标准的 `httpd`端口是80。一般情况下,应该与标准端口相连,这样可以使人们很容易进入

你提供的服务。于是这一行应为：

```
Port 80
```

```
3) # HostnameLookups
```

这个选项的缺省为 HostnameLookups off，这样的缺省对所有的用户都有好处，当然也可以设置为 on，如：www.apache.com on 210.45.76.6 off

```
4) #User nobody
```

```
#Group nobody
```

如果希望你的 httpd 对不同的用户和群组有不同的操作（通常不需要这么做），那么开启这个开关，若无此要求，则设置为：

```
User nobody
```

```
Group nobody
```

```
5) # ServerAdmin: Your address, where problems with the server should be e-mailed.
```

当有人向你写信提及有关提供的服务时，不必紧张。能听到正面的评论是很好的，但当有人提出建设性的意见批评时，说明你的站点所提供的服务还需要进一步完善。因此，此处含有负责维护 Web 服务的 E-mail 地址。笔者所看到的是：

```
ServerAdmin root@localhost
```

```
6) # ServerRoot: The directory the server's config, error, and log files are kept in.
```

这里告诉你的系统在何处可以找到 httpd 文件。当然，将它们放在何处完全是个人爱好问题。

```
ServerRoot /etc/httpd
```

7) 现在告诉系统在何处可以找到 httpd 日志文件。假如你的路径不是以 / 开头，意味着开始目录由 ServerRoot 分配。日志文件如下：

- ErrorLog 跟踪超时客户，不产生输出的脚本，试图覆盖不允许覆盖的内容，严重的服务器错误（产生段错误或总线错误）以及用户权限有关的问题。
- TransferLog 跟踪以下信息：客户主机，提供的 identd 信息，用户权限，进入的日期和时间，HTTP 请求的第一行，服务器状态码，以及发送的字节总数。
- AgentLog 跟踪使用的客户代理软件，它大多数情况用于收集统计数字或跟踪导致服务器错误的客户。
- RefererLog 跟踪指导用户与你连接的那些文档。
- 与这个日志文件相关的一个变量为 ReferLgnore，它忽略与特定用户的连接。例如，当不想对用户每次翻阅你的页面时都做记录，就可以使用它。因为笔者建立了 www.renaissoft.com，所以选择忽略那个主机。
- PidFile 跟踪你的服务器进程 ID 号。
- 可以选择在某类中央记录区保存你的记录，若决定将我的记录与其他的 httpd 内容保存在一起，配置文件中的其他项将为：

```
ErrorLog logs/error_log
```

```
CustomLog logs/access_log common
```

```
PidFile /var/run/httpd.pid
```

```
ScoreBoardFile /var/run/httpd.scoreboard
```

8) #StartServers: The number of server to launch at startup

尽管对处理服务器数目有一编译缺省值，最好自己设置以了解你所要使用的数目。假如你接受这一缺省值，则很难知道现在可用的服务器，除非你查原代码。笔者选择的与例子相同的服务器数目(10)，这样我的系统就可以同时处理10个HTTP请求，而不需要额外的资源来启动多出的httpd进程。同时，10个空闲的daemon只可能占用少量的资源。因此，这一行为：

```
StartSever10
```

9) #MaxSpareServer:

#MinSpareServer:

同样，有一编译缺省值，但建议在你的配置文件中设定一下。上例中数目为20和10，即最大和最小空闲服务器进程数目。系统将周期性检测等待中的服务进程数目，如果数目小于最小空闲服务器进程数目，系统将会产生一个新的进程；若数目大于最大服务器进程，将是一种资源的浪费，系统将会关闭一个进程。笔者将其设置为：

```
MinSpareServers 8
```

```
MaxSpareServers 20
```

10) # MaxKeepAliveRequests

这个参数表明在当前的连接中系统能够处理的请求的最大数值，为了保证你的服务器正常服务，这个参数应该设置得较大。笔者在模板文件中的参数设置为：

```
MaxKeepAliveRequests 100
```

11) MaxClients 150

这个参数值表示服务器允许的最大客户请求数，即最多允许访问用户的数目。

12) # MaxRequestsPerChild

MaxRequestsPerChild 这个指令设定一个独立的子服务程序将能处理的请求数量。在处理MaxRequestsPerChild这个请求之后，子程序将会被终止。如果MaxRequestsPerChild为0的话，那么该程序永远不会被终止。MaxRequestsPerChild设为非0的值做为限制有两个好处：

- 由记忆体使用量限制程序使用的记忆体数量；
- 由赋予程序有限的存留时间，可以在服务器负载降低时协助减少程序的数目。

3. 建立资源配置文件

建立这个文件用于通知Web服务器从哪里可以找到文档、脚本和别名。下面是一个名为etc/httpd/conf/srm.conf的文件，作为模板。供大家参考。

```
# DocumentRoot: The directory out of which you will serve your
# documents.
DocumentRoot /home/httpd/html
# UserDir: The name of the directory which is appended onto a user's home
# directory if a ~user request is recieved.
UserDir public_html
# DirectoryIndex: Name of the file or files to use as a pre-written HTML
# directory index. Separate multiple entries with spaces.
DirectoryIndex index.html index.shtml index.cgi index.htm
# FancyIndexing is whether you want fancy directory indexing or standard
FancyIndexing on
# AddIcon tells the server which icon to show for different files or filename
# extensions
```

```
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*
AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
# DefaultIcon is which icon to show for files which do not have an icon
# explicitly set.
DefaultIcon /icons/unknown.gif
# HeaderName is the name of a file which should be prepended to
# directory indexes.
ReadmeName README
HeaderName HEADER
# IndexIgnore is a set of filenames which directory indexing should ignore
# Format: IndexIgnore name1 name2...
IndexIgnore .??* *~ *# HEADER* README* RCS
# AccessFileName: The name of the file to look for in each directory
# for access control information.
AccessFileName .htaccess
# TypesConfig describes where the mime.types file (or equivalent) is
# to be found.
TypesConfig /etc/mime.types
# DefaultType is the default MIME type for documents which the server
# cannot find the type of from filename extensions.
DefaultType text/plain
# AddEncoding allows you to have certain browsers (Mosaic/X 2.1+) uncompress
# information on the fly. Note: Not all browsers support this.
AddEncoding x-compress Z
AddEncoding x-gzip gz
# Just list the languages in decreasing order of preference.
LanguagePriority en fr de
# Redirect allows you to tell clients about documents which used to exist in
```

```
# your server's namespace, but do not anymore. This allows you to tell the
# clients where to look for the relocated document.
# example.
Alias /icons/ /home/httpd/icons/
# ScriptAlias: This controls which directories contain server scripts.
# Format: ScriptAlias fakename realname
ScriptAlias /cgi-bin/ /home/httpd/cgi-bin/
# If you want to use server side includes, or CGI outside
# ScriptAliased directories, uncomment the following lines.
# AddType allows you to tweak mime.types without actually editing it, or to
# make certain files to be certain types.
# Format: AddType type/subtype ext1
Addtype application/x-httpd-cgi .pl .cgi
#AddType application/x-httpd-php3 .php3
#AddType application/x-httpd-php3-source .phps
# The following is for PHP/FI (PHP2):
#AddType application/x-httpd-php .phtml
# AddHandler allows you to map certain file extensions to "handlers",
# actions unrelated to filetype. These can be either built into the server
# or added with the Action command (see below)
# Format: AddHandler action-name ext1
# To use CGI scripts:
AddHandler cgi-script .cgi
# To use server-parsed HTML files
AddType text/html .shtml
AddHandler server-parsed .shtml
# Uncomment the following line to enable Apache's send-asis HTTP file
# feature
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0
```

1) # DocumentRoot: The directory out of which you will serve your
documents.

使用这一变量，可以规定你的服务器将显示文件的主目录。通常，笔者使用标准方法（如果你 想将所有的记录文件放在一个集中的地方，可以改变它，以适应你的需要），这样，此文件的开头为：

```
DocumentRoot /home/httpd/html
```

2) # UserDir: The name of the directory which is appended onto a user's home
directory if a ~ user request is recieved.

这个指令用于分配一个目录，此目录将存在于用户保存他们的 HTML文件的目录中，标准是目录：public_html: 任何时候如果某人跟踪某个链接，例如 <http://www.sever.com/~user>，他实际进入的目录应该是：http://www.sever.com/~user/public_html，这是一种保持整洁及适当的措施，这样其他人员便不能进入用户目录。

笔者的配置文件的下一行为(使用了缺省值)

```
UserDir public_html
```

3) # DirectoryIndex: Name of the file or files to use as a pre-written HTML
directory index.

使用这一指令，当外面的用户连接到某人的主目录时，可以设置一个假想的文件名。在上例中，并没有列出文件名，只是因为已设置了缺省的文件名，于是服务器首先查看此文件名(一般为index.html或index.htm)。笔者将其设置为

```
DirectoryIndex index.html index.shtml index.cgi index.htm
```

```
4) # FancyIndexing is whether you want fancy directory indexing or standard
```

FancyIndexing指显示目录、文件及图标和文件大小。假如你想以此方式显示你的文件和目录，将这一选项打开；否则，将其关闭。笔者的下一行设置为 On，形式如下：

```
FancyIndexing on
```

```
5) # AddIcon tells the server which icon to show for different files or filename
```

```
# extensions
```

可以使用这个指令定义 Web服务器将使用什么图标显示目录中的文件类型。这是一种改进目录显示方法的简单方法。这个指令没有缺省值，因此如果你没有定义内容，就没有图标可用。

欲使用图标的路径蕴含在 DocumentRoot变量的值中，而 DocumentRoot变量在这个配置文件的一开始就已定义了，因此，第一个变量指出了保存图标的目录，这个目录将在你的 DocumentRoot目录内。笔者的配置如下：

```
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
```

```
AddIconByType (TXT,/icons/text.gif) text/*
```

```
AddIconByType (IMG,/icons/image2.gif) image/*
```

```
AddIconByType (SND,/icons/sound2.gif) audio/*
```

```
AddIconByType (VID,/icons/movie.gif) video/*
```

```
AddIcon /icons/binary.gif .bin .exe
```

```
AddIcon /icons/binhex.gif .hqx
```

```
AddIcon /icons/tar.gif .tar
```

```
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
```

```
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
```

```
AddIcon /icons/a.gif .ps .ai .eps
```

```
AddIcon /icons/layout.gif .html .shtml .htm .pdf
```

```
AddIcon /icons/text.gif .txt
```

```
AddIcon /icons/c.gif .c
```

```
AddIcon /icons/p.gif .pl .py
```

```
AddIcon /icons/f.gif .for
```

```
AddIcon /icons/dvi.gif .dvi
```

```
AddIcon /icons/uuencoded.gif .uu
```

```
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
```

```
AddIcon /icons/tex.gif .tex
```

```
AddIcon /icons/bomb.gif core
```

```
AddIcon /icons/back.gif ..
```

```
AddIcon /icons/hand.right.gif README
```

```
AddIcon /icons/folder.gif ^^DIRECTORY^^
```

```
AddIcon /icons/blank.gif ^^BLANKICON^^
```

```
6) # DefaultIcon is which icon to show for files which do not have an icon
```

```
# explicitly set.
```

这里，告诉服务器用哪个图标表示未曾定义的文件类型。一般为 unknown图标。笔者使用缺省配置，其文件的下一行为：

```
DefaultIcon /icons/unknown.gif
```

7) #ReadmeName is the name of the README file the server will look for by
default.

使用此指令定义 HTML 文件，它包含一个当前目录下的文件描述列表。尽管文件名与描述分别储存在不同的文件中，但这些文件描述在目录列表的文件旁边显示。

以后你将需要创建包含描述的 HTML 文件。为了简单起见，起名为 readme.html，这样源配置文件的下行为：

```
ReadmeName README
```

8) # HeaderName is the name of a file which should be prepended to
directory indexes.

使用此指令定义包含一个当前目录自定义标题的 HTML 文件，这个标题在目录列表的上部显示。今后你将有必要创建包含最新标题的 HTML 文件。为了简单起见，我将它定义为 header.html，于是配置文件的此行为：

```
HeaderName HEADER
```

9) # IndexIgnore is a set of filenames which directory indexing should ignore
Format: IndexIgnore name1 name2...

假如不想让某些文件出现在目录列表中，使用 IndexIgnore。例如，假如你想忽视 readme.html 及 header.html 文件（它们由服务器本身使用，用户无需亲自使用它们），笔者将使用下一行：

```
IndexIgnore .??* *~ *# HEADER* README* RCS
```

10) # AccessFileName: The name of the file to look for in each directory
for access control information.

这条指令告诉服务器在何处去寻找访问控制文件，你将对这种文件进行配置，确立哪些目录是高于一般客户的人才能进入的。

```
AccessFileName .htaccess
```

11) # DefaultType is the default MIME type for documents which the server
cannot find the type of from filename extensions.

当服务器不能决定实际文件类型时，使用这一指令可以分配一个缺省的文件类型。我使用的缺省设置：

```
DefaultType text/plain
```

这里，text 指文件类型（标准文件类型），plain 指子类型（MIME 调用这种文件类型）。

12) # AddEncoding allows you to have certain browsers (Mosaic/X 2.1+) uncompress
information on the fly. Note: Not all browsers support this.

假如想使你的服务器在发送文件的同时对其解压缩（这样存储该文件的所需的磁盘空间较少），可使用这个指令。例如，若想让服务器能够还原 Unix 压缩 (.z) 文件及 gzip 压缩 (.gz) 文件（正如源配置文件列表中的例子所示），将输出下述内容：

```
AddEncoding x-compress Z
```

```
AddEncoding x-gzip gz
```

13) # AddLanguage allows you to specify the language of a document. You can
then use content negotiation to give a browser a file in a language

```
# it can understand.
```

此项允许你指定文档所使用的语言，可以使用缺省值。其文件如下：

```
AddLanguage en .en
AddLanguage fr .fr
AddLanguage de .de
AddLanguage it .it
```

其中en表示英语，fr为法语，de为德语，it则为意大利语。随后的配置 LanguagePriority与此项类似，这里从略。

14) # Aliases: Add here as many aliases as you need (with no limit). The format is

```
# Alias fakename realname
```

假如想为你的HTML文件路径起一个别名，可以使用这个指令。例如，将/home/httpd/icons/起别名为/icons,可以用如下的指令：

```
Alias /icons/ /home/httpd/icons/
```

15) # AddType allows you to tweak mime.types without actually editing it, or to

```
# make certain files to be certain types.
```

如果希望你的服务器能够支持 CGI(大多数的Web服务器都这样做了)，利用此选项，把它加入你的配置文件，笔者的下一行为：

```
Addtype application/x-httpd-cgi .pl .cgi
```

同样，如果希望你的服务器同样支持 php,也可以以同样的方法将其相加，如：

```
AddType application/x-httpd-php .phtml
```

16) # AddHandler allows you to map certain file extensions to "handlers",

```
# actions unrelated to filetype.
```

4. 建立访问配置文件

这是要设置的最后一个配置文件，它告诉 Web服务器哪些客户可以进入哪些目录，以及在其他目录哪些服务是可用的。下面的文件为 etc/httpd/conf/access.conf，可作为大家配置此文件的模板。

```
## access.conf -- Apache HTTP server configuration file
# access.conf: Global access configuration
<Directory />
Options None
AllowOverride None
</Directory>
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something working as
# you might expect, make sure that you have specifically enabled it
# below.
# This should be changed to whatever you set DocumentRoot to.
<Directory /home/httpd/html>
# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
Options Indexes Includes FollowSymLinks
# This controls which options the .htaccess files in directories can
```

```
# override. Can also be "All", or any combination of "Options", "FileInfo",
# "AuthConfig", and "Limit"
AllowOverride None
# Controls who can get stuff from this server.
order allow,deny
allow from all
</Directory>

# /home/httpd/cgi-bin should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
<Directory /home/httpd/cgi-bin>
AllowOverride None
Options ExecCGI
</Directory>

<Directory /home/httpd/html/hbv>
Options ExecCGI
</Directory>

<Directory /home/httpd/html/ipv6board/cgi-bin>
Options ExecCGI
</Directory>

<Directory /home/zam/public_html/cgi/>
Options ExecCGI
</Directory>

# Allow server status reports, with the URL of http://servername/server-status
# Change the ".your_domain.com" to match your domain to enable.
#<Location /server-status>
#SetHandler server-status
#order deny,allow
#deny from all
#allow from, your_domain.com
#</Location>

# Allow access to local system documentation from localhost
Alias /doc /usr/doc
<Directory /usr/doc>
order deny,allow
deny from all
allow from localhost, 127.0.0.1
Options Indexes FollowSymLinks
</Directory>

# There have been reports of people trying to abuse an old bug from pre-1.1
# days. This bug involved a CGI script distributed as a part of Apache.
# By uncommenting these lines you can redirect these attacks to a logging
# script on phf.apache.org. Or, you can record them yourself, using the
script
# support/phf_abuse_log.cgi.
#<Location /cgi-bin/phf*>
#deny from all
#ErrorDocument 403 http://phf.apache.org/phf_abuse_log.cgi
#</Location>

# You may place any other directories or locations you wish to have
# access information for after this one.
```

要建立你自己的配置文件etc/httpd/conf/access.conf，可照上例中同样的顺序填写内容。从

某种意义上讲，这是最重要的一个配置文件，因为它控制对你的系统的存取，所以要多花些时间学习，下面是我的服务器配置文件的一些选择。

```
1) # This file defines server settings which affect which types of services
# are allowed, and in what circumstances.
<Directory />
Options None
AllowOverride None
</Directory>
# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".
```

从上面可以看出，服务器在此目录下的选项如下：

- None 表示在此目录下没有特性，这是最安全的，因为使用你服务器的用户完全不能在此目录下做任何事情。
- All 表明此目录下的所有特性可用。不要轻易使用这个特性，除非你真的想让用户进入特定的目录看到其中的任何东西（包括用户的主目录）。实际上，假如从一个用户的主目录到你的某个目录建立了 simlinks,而这个目录你并不想让其他人进入。如果使用 All 选项，其他的用户就可以利用这个链进入该目录。
- FollowSymLinks 使得服务器从这个目录跟踪 simlinks。选择此项一定要小心，因为如果 simlinks 是从用户的主目录到不允许其他人访问的目录，将会产生一种链接，使得外部的 Web 用户也可以进入你并不想让他进入的目录。
- SymLinksIfOwnerMatch 使得服务器从这个目录跟踪 simlinks，条件是链所指的文件拥有者同时是链的创建者。此选项部分地解决了上面所述的问题，外部用户一般不能跟踪链，除非该用户也拥有此链所连接的文件。
- Indexs 表明服务器可以让人们要求当前目录的索引，并让部分用户得到他们所在目录的 HTML 索引。如果不用此选项，外部用户将不会得到 HTML 索引，但在此目录中可以加载任何要索引的文件。如果想提供 HTML 文件索引及描述，使用此选项。
- Include 表明服务器可以让用户提供信息，例如一个文件最后被修改的日期。

在这里，出于安全起见，先将所有的目录都定义为 None，以免出现安全漏洞。由于此选项是可以覆盖的，需要其他要求的选项可以在后面一一定义。

```
2) # This should be changed to whatever you set DocumentRoot to.
<Directory /home/httpd/html>
Options All
AllowOverride All
order allow,deny
allow from all
</Directory>
```

这个 Directory 指令同样也是一个完整的语句。和以前一样的模式，但这一次指的是 SeverRoot 的目录（我使用的是 home/httpd/html），此部分的第一行为：

```
<Directory /home/httpd/html>
```

此时，我作为根目录又完全控制了该目录，但想要让人们能够获得这里所有的文件，可以使用一下选项。

```
Options All
```

现在已可以通过输入来控制哪个 .htaccess 选项是服务器可以覆盖的。尽管有多种选择，我建议最好用下面的：

```
AllowOverride All
```

除非你确实想拒绝一些主机存取，一般情况下需要做的仅仅是添加为以下部分：

```
allow from all
```

```
3) # /home/httpd/cgi-bin should be changed to whatever your ScriptAliased
```

```
# CGI directory exists, if you have that configured.
```

```
<Directory /home/httpd/cgi-bin>
```

```
AllowOverride None
```

```
Options ExecCGI
```

```
</Directory>
```

以上为 SeverRoot 的设置，笔者采用了缺省值，其选项如上所示。

4) 现在设置 usr 目录下的文件特性，笔者选择如下：

```
#Allow access to local system documentation from localhost
```

```
Alias /doc /usr/doc
```

```
<Directory /usr/doc>
```

```
order deny,allow
```

```
deny from all
```

```
allow from localhost, 127.0.0.1
```

```
Options Indexes FollowSymLinks
```

```
</Directory>
```

其中，笔者只允许内部 IP 127.0.0.1 的主机查看这个目录，因此做了以下的设置：

```
deny from all
```

```
allow from localhost, 127.0.0.1
```

到此为止，你已经完成了配置 Web 服务器的工作！

5. 启动 Web 服务器

这是最后一步，真正地启动 Web 服务器。设置一些内容，以便每次引导系统时服务器都将自己启动。需要做以下的工作：

1) 编辑文件 /etc/service。

2) 增加以下行：

```
http 80/tcp
```

你不必使用端口 80，但它是标准的 http 端口，若不使用这个端口，人们将不得不在 URL 中使用端口编号才能与你连接。

3) 存盘退出 /etc/services。

4) 编辑文件 /etc/inetd.conf。

增加行：

```
http stream tcp nowait ftp /etc/httpd/httpd httpd
```

5) 存盘退出。

6) 作为 root 登录。

7) 输入 ps aux。

8) 为 inetd 找到进程号。

9) 输入 kill <process#>- HUP

从现在开始，当你引导系统时，自动启动的一个进程便是 httpd。

6.4 建立Samba服务器

不管现在身在何处，只要身边有电脑的话，几乎都可以看到 Windows95/NT的踪影，因为 Windows95/NT的GUI界面漂亮、直观、简单易用，不能不让人想使用它，但 Linux的爱好者喜欢Linux系统的开放和每天有挖掘不完的新东西，自己可以把 Linux弄的千变万化，完全适用于自己的电脑与本身的喜好。两种操作系统的优良特性如果能互相配合，那不是挺好的吗？但是Linux与Windows95/NT之间的连线只能靠TCP/IP，从Windows95/NT telnet到Linux或是用ftp做文件传输而已！Linux虽然有NFS可以有效和Windows95/NT相连，然而Windows95/NT目前似乎没有一套免费、易用的NFS软件可以和Linux沟通。两种操作系统之间似乎毫无关系！但自从“Samba”出现后，从此为Linux和Windows95/NT搭起友谊的桥梁了。

6.4.1 Samba是什么

基本上，Samba可以想成是一个局域网络上的File/Printer Server，可以提供文件系统、打印机或是其他的信息并与Samba Server在同一个子网域(subnet)的Samba client(如Windows95、WindowsNT、WarpServer、smbfs等)共享。因为它的基本工作原理是让Netbios(Windows95网络邻居的通信协议)与SMA(Server Message Block)这两种协议运行在TCP/IP的通信协议上，且使用Netbios nameserver让你的Linux机器可以在网络邻居上被看到，所以Linux就可以和Windows95/NT在网络邻居上沟通，互相浏览分享文件了。也就是说，Samba Server可以让你的Linux变成如Novell Server一般，可以让Windows95的使用者用网络邻居分享Linux的文件和打印机了。

6.4.2 安装Samba

Samba这套免费的软件里面包含了Samba Server和Samba Client，所以装好Samba后，不只是可以提供资源给别人，同样也可以用Samba Client去分享其他Samba Server或是Windows95/NT网络邻居上的机器共享出来的资源。

基本上，Samba是为UNIX系统所发展的，但目前也有NetWare、Windows NT、OS/2和VMS版的Samba了，如果想尝试使用别的操作系统的Samba也是非常容易的一件事！而在使用Red Hat Linux安装Samba之前，建议kernel最好是用2.2.10版的，因为它针对Samba的bug做了些修正，在使用Samba client打印文件名时会稳定一些。

在Red Hat Linux下安装Samba是最简单不过的了，只要几个步骤就可以解决了。目前最新版的Samba版本为2.0.3p8，表示2.0.3版patch第8次。在Red Hat Linux发行版所附光盘中，即捆绑有Samba-2.0.3-8.i386.rpm软件，你可以直接从光盘上安装。当然了，你也可以从其他渠道(比如Internet上Samba的主页：<http://www.apache.org>上下载Apache的最新版本)获得Samba软件，如果是Samba的源程序代码压缩文件，你首先得将它进行解压缩，然后进行编译才可以使用。

以Samba-1.9.17p2.tar.gz文件为例，讲一下编译的基本步骤：(如果使用Red Hat Linux 4.1光盘中捆绑的Samba-2.0.3-8.i386.rpm软件，因为它是RPM类型文件，无须编译，可直接跳过这一步骤)。

我们现在就开始安装Samba。请依照下列步骤完成：

1) 解压缩文件。

取得samba-1.9.17p2.tar.gz文件，然后随便找个目录将它解压缩，包文件解压缩后，直接会产生目录samba-1.9.17p2/：

```
# tar zxvf samba-1.9.17p2.tar.gz
```

阅读安装说明文件，获取帮助，先看看 README，再到docs/下看看其他文件和执行：

```
# nroff -man smbd.8|less
```

```
# nroff -man nmbd.8|less
```

相信会得到非常多的帮助。

2) 进入source/，编辑Makefile文件。

先看一下README、INSTALL之类的文件，对于安装软件会有很大的帮助。因为 Samba 可适用于多种UNIX操作系统，所以 Makefile里面定义了非常多的UNIX系统，我们只要找出定义Linux的部份，然后将#删除就可以了。而这一版 Samba的Makefile里面对Linux有五种定义情况，请选择一种。基于安全性考虑，建议用第三种方式包含有 shadow password与quota来编译Samba：

```
Use this for Linux with shadow passwords
```

先从Makefile中找到上面这一段字。假如系统有安装 shadow password，就将Makefile里面这一段的

```
# FLAGSM = -DLINUX -DSHADOW_PWD -DFAST_SHARE_MODES
```

```
# LIBSM = -lshadow
```

这两行的#号删除，然后储存Makefile进行第3步骤。

```
Use this for Linux without shadow passwords
```

假如你的系统没有安装 shadow password，就将Makefile里面这一段的

```
# FLAGSM = -DLINUX -DAXPROC -DFAST_SHARE_MODES
```

```
# FLAGSM = -DLINUX -DFAST_SHARE_MODES
```

```
# LIBSM =
```

这三行的#号删除，然后储存Makefile进行第3步骤。

```
Use this for Linux with shadow passwords and quota
```

假如系统有安装 shadow password和quota的话，就将Makefile里面这一段

```
# FLAGSM = -O3 -m486 -DLINUX-DSHADOW_PWD -DQUOTAS -DFAST_SHARE_MODES
```

```
# LIBSM = -lshadow
```

这两行的#号删除，将Makefile储存起来后，并直接在source/这个目录执行：

```
ln -s /usr/include/linux/quota.h /usr/include/sys/quota.h
```

或者

```
ln -sf /usr/src/quota-1.51 quota
```

如果/usr/src/目录下没有 quota-1.51，可以到光盘的 quota/目录下取得 quota-1.51.tar.gz，然后将它解开再 link即可。多做这两个步骤的原因是因为要包含 quota，所以编译时需要用到 quota的source。

```
Use this for Linux with quota and glibc2
```

glibc2是GNU新开出的C Library，所以若有使用glibc2与quota，请将下面选项的#号删除：

```
# FLAGSM = -DLINUX -DQUOTAS -DNO_ASMSIGNALH -DGLIBC2 -DFAST_SHARE_MODES
```

```
# LIBSM = -L/lib/libc5-compat-1crypt
Use this for Linux with glibc2(libc6)
```

只有使用glibc2的人，请将下面选项的#删除：

```
# F L A G S M - B L I N U - X D N E T G R O - U D P S H A D O W _ P W A D L L O W _ C H A N G E _ P A S S W C
-DFAST_SHARE_MODES = 1 -DNO_ASMSIGNALH_DGLIBC2
# LIBSM = -lnsl -1crypt
```

3) 在source/执行“make”

make成功后会得到好几个可执行文件，但我们执行 Samba Server或client最需要的应是“smbd”、“nmbd”这两个Samba的守护进程和“smbclient”这个Samba的客户进程的执行文件并生成。接着执行“make install”，会将Samba的所有东西安装到/usr/local/samba/这个目录，就完成安装samba了。若以前装过Samba，可以执行“make revert”将旧的版本删除并安装新版本的Samba。

```
# make
# make install(or make revert)
```

做完此3个步骤就完成编译httpd了。

以上只是编译Samba源代码的一般步骤，在RedhatLinux中，根本不必如此麻烦，可以直接使用RedhatLinux光盘中捆绑的Samba-2.0.3-8.i386.rpm软件，因为它是RPM类型文件，无须编译即可直接安装。

首先将光盘放入CDROM中，可以执行

```
$ mount
$ cd /mnt/cdrom/
```

确认CDROM已经Mount上，这时可以从光盘的/RedhatLinux/RPMS/下取得Samba-2.0.3-8.i386.rpm这个rpm软件，然后用RPM将它解压缩进行安装，RPM这个安装软件可以将新软件的source code重新整理包装成另外一种source或binary的文件型式(.rpm)，这样一来可以免去安装软件时编译发生错误所产生的不良后果。(RPM软件的详细介绍在本章最后一节。)执行命令如下：

```
$ rpm -ivh Samba-2.0.3-8.i386.rpm
Samba#####
```

可以看出Samba的软件名称以及它的安装过程，等到#字符跑完，就是软件安装完成了。安装完成后会自动在/etc/这个目录下放置有产生Samba的samba.conf配置文件，在/var/log/samba/下面放置log.nmb文件，下面我们将详细谈如何进行设置。

6.4.3 设置Samba的配置文件

Samba的配置文件smb.conf关系着Samba Server开放的权限、目录、打印机和机器所属的组，所以它对Samba Server非常重要。在使用RedhatLinux光盘自带的samba压缩文件samba-2.0.3-8.i386.rpm进行解压后，在example/simple/目录下就有一个smb.conf的范例，请将它复制到/usr/local/samba/lib/这个Samba预设读取smb.conf的目录，另外/etc/下有一个smb.conf.default，这是完整的smb.conf说明使用文件，也一并把它复制到/usr/local/samba/lib/。接着我们开始修改smb.conf。

为安全起见，我们首先还是把/etc/smb.conf文件复制成smb.confold，smb.conf文件，

其内容说明如下：

smb.conf范例说明：

打开smb.conf文件，首先是如下一段说明文字，建议你在明白设置的含义后进行修改。这是SambaServer---smbd的设置文件；请特别注意，smb.conf文件把用分号“；” mark起来的叙述当作注解。

```
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options (perhaps too
# many!) most of which are not shown in this example
#
# Any line which starts with a ; (semi-colon) or a # (hash)
# is a comment and is ignored. In this example we will use a #
# for commentry and a ; for parts of the config file that you
# may wish to enable
#
```

下面的workgroup这一行用来指定你的机器在网络邻居上所属的组名称。笔者定义为 qc。如果不指定，则内置的组名称为：“MYGROUP”

```
workgroup = qc
```

下面的hostsallow这一行是设置哪一些领域的机器可以访问你的 Samba Server；若没有设置，则内置任何机器都可以访问你的 Samba；这里是设置 192.168.1. 192.168.2. 127.领域的机器才可以访问SambaServer。

```
hosts allow = 202.204.89. 202.204.89. 120.
```

下面的hostsallow这一行是定义如果你想使用自动调用的打印机而不是当需要时自己设置。

```
printcap name = /etc/printcap
load printers = yes
```

下面的config file这一行是设置，只要加上下面这一行，不用重新开机，就可让修改smb.conf后马上生效。不过对Workgroup的定义无效。

```
config file = /etc/smb.conf
```

下面是SamabServer的一般化设置，设置打印机(须配合；/etc/printcap设置)，使用者帐号为guest，将log file文件(记录文件)设置放在/var/log/samba/log.%m目录下；share modes = yes指定允许任何人(guest)使用定义在/etc/printcap下的打印机。

```
[global]
; printing = bsd
; guest account = pcguest
log file = /var/log/samba/log.%m
max log size = 50
share modes = yes
```

下面定义使用SamabServer的用户名的目录。

```
; username map = /etc/smbusers
```

下面是关于linux的printer的使用设置，设置打印机在 Windows95网络邻居上的名称为[激光打印机]，且这台打印机可以在网络邻居上看到(browseable = yes)，可以执行打印(printable = yes)；但无权限者，如guest不可以使用(public = no)，且这台打印机的权限设置为 0750，也

就是只有拥有者(root)才能使用。

```
[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
# Set public = yes to allow users to print
guest ok = no
writable = no
printable = yes
printable = no
create mask = 0750
```

下面设置所有使用者的Home Directories：

```
[homes]
comment = Home Directories
browseable = yes
path = /home
writable = yes
```

下面设置david这位linux上的使用者的Home directory。这种设置也是安装samab的好处之一，david这位使用者可以直接经由Windows95的网络邻居存取它在linux上的Home directory，对使用者来讲非常方便。

```
[david]
comment = david's Home
path = /usr/david
valid users = david
public = no
writable = yes
```

下面的语句设置一个暂存的目录，可以让任何人共享 (public = yes)：

```
[tmp]
comment = Temporary file space
path = /
read only = yes
public = yes
```

上面只举了几个范例，如果想知道更多的范例，请参考 Samba内附的smb.conf . default。设置好smb.conf后，紧接着就要启动Samba Server了。

6.4.4 启动Samba

启动Samba Server有两种方法，一种是以stand-alone的形式，另一种则是以Inetd的形式。这两种方式各有利弊，以stand-alone的方式运行，Daemon会随时占住系统资源，但反应速度较快，所以若没有电脑资源分配的问题，笔者建议采取这种方式。而以Inetd的方式运行则比较省电脑资源，相对的执行速度会比较慢。选择哪种方式，完全根据自己机器的设置而定。

要特别注意的是，不管采用哪一种方式启动Samba都可以，但切记不要让两种方式同时执行，否则Samba Server将无法正常运行！然而，安装Samba时，Samba已经“预设使用Inetd的形式”启动Samba，所以此时只要重新开机后，Samba Server就会启动了。如果要使用Daemon启动Samba，则必须先删除Inetd形式启动的设置，所以我们先来了解如何用Inetd启动Samab吧！

1. 使用Inetd启动Samab

(1) 在 “ /etc/services ” 文件中加入下面六行

```
netbios-ns 137/tcp nbns
netbios-ns 137/udp nbns
netbios-dgm 138/tcp nbdgm
netbios-dgm 138/udp nbdgm
netbios-ssn 139/tcp nbssn
netbios-ssn 139/udp nbssn
```

因为安装Samba时已经自动帮加了，所以不用自己动手，如果你准备用 stand-alone的形式启动，就在这六行的前面加上#，将他们注释起来吧！

(2) 在 “ /etc/inetd.conf ” 文件中加入下面两行

```
netbios-ssn stream tcp nowait root /usr/sbin/smbd smbd
netbios-ns dgram udp wait root /usr/sbin/nmbd nmbd
```

并请将在 /usr/local/samba/bin/下的smbd、nmbd这两个Samba daemon文件复制到/usr/sbin/以方便使用。

2. 以stand-alone的方式启动Samba

以stand-alone的方式启动Samba在RedhatLinux下最简单不过了，只要将下面三行加入 “ /etc/rc.d/rc.local ” 这个文件中就可以了：

第一行是用echo显示信息用的，不加也没关系，接着是执行 smbd和nmbd。而参数-D就是表示要以 Daemon形式运行，-d1就是设置除错的级别为第一级，总共有 0~5共六级。至于nmbd后面的-n samba就是指定这一台机器在 Windows 95网络邻居上看到的名称为 “ samba ”，你可以自己改成自己喜欢的名称，再配合 smb.conf 文件中定义的workgroup=Samba House，也就是说在网络邻居的 Samba.House里面会有一台机器名称叫做 samba。

如果说Samba启动都完全没问题，Linux和Windows 95之间的网络也都通畅的话，则可以在网络邻居上看到在整个网络上 Samba_House这个组名称。这个组名称里面有一台叫 Samba的机器，不知道的人，在 Windows 95的网络邻居看到你，还以为你的电脑也是安装 Windows 95呢？假如出现Samba无法启动情况，请检查 /usr/log/samba/目录下的log.nmb(nmbd启动记录)与log.smb(smbd启动记录)，以修正问题。

6.4.5 Samba常用的重要指令

当你的Samba运行正常时，请也看看这一小节，因为会介绍许多 Samba常用的重要指令。如果你的Samba Server不能正常工作，请依照下列步骤完成测试工作：

首先至少要有两台电脑可以使用，一台是你的 Linux，一台则是Windows 95的机器，这样才能进行测试工作。另外可以将 /usr/local/samba/bin/这个路径加入 .profile或.cshrc中(假如使用bash或tcsh的话)，以方便使用Samba的执行程序。

1. Test1

使用Samba提供的测试程序 “ testparm ”，测试smb.conf的设置是否正确。

```
# testparm
```

2. Test2

使用ping，分别从Linux和Windows 95互相ping对方，如果双方都有反应的话，表示网络

是顺畅的。

3. Test3

在Linux上执行“`smbclient -L hostname`”，测试能不能列出Samba Server共享出来的资源。如笔者的机器hostname为david，所以执行：

```
# smbclient -L david
```

会列出类似下面的信息

```
Added interface ip=140.115.210.158 bcast=140.115.255.255 nmask=255.255.0.0
Server time is Sun Oct 06:26:30 1999
Timezone is UTC+80
Domain=[SAMBA_HOUSE] OS=[Unix] Server=[Samba 1.9.16p10]
Server=[david] User=[root] Workgroup=[SAMBA_HOUSE] Domain=[SAMBA_HOUSE]
SharenameTypeComment
HomesDiskHome Directories
IPC$IPCIPC Service (Samba 1.9.16p10)
RootDiskHome Directories
wzyangDiskwzyang'sHome
激光打印机Printer All Printers
暂存区 Disk Temporary file space
```

如果可以列出类似这样的信息，Samba Server应该没什么问题了！而这里要特别介绍的就是smbclient这个程序，smbclient是Samba提供的client程序，使用它可以存取网络上任何一台Samba Server(Windows 95/NT网络邻居上的任何一台PC也算在内)开放出来的文件。而我们在该测试中加-L就是列出Samba Server提供出来的资源清单。

4. Test 4

在Linux上执行“`nmblookup -B hostname_SAMBA`”，应该会得到Samba Server回传你的机器的IP Address。例如：

```
# nmblookup -B david_SAMBA_
```

假如没有回传的话，而且是用inetd启动Samba的话，则请检查inetd.conf中关于nmbd的设置是否正确。如果使用standalone启动的话，可能是在smbd或nmbd的地方加错参数了。

5. Test 5

在Linux上执行“`nmblookup -B IP_Address*'`”，其中的IP_Address是你使用的Windows 95机器的网址，假如我使用的Windows 95其IP为140.115.219.145，则执行：

```
# nmblookup -B 140.115.219.145
```

则Windows 95那一台机器应该要回传它的IP回来。如果没有的话，应该是IP address不正确，或是Windows 95没开机，不然就是Windows 95关于网络的设置没有安装好。

6. Test 6

在Linux上执行“`nmblookup -d 2*'`”，会回传和Samba Server在同一个子网域(subnet)的所有Windows 95机器网址。类似Got a positive name query response from 140.115.210.97这样的信息。通常Test 5没问题的话，这里应该也没问题。

7. Test 7

在Linux上执行“`smbclient \\hostname\Sharename`”，登录自己的Samba Server。hostname就是表示你的机器名称，Sharename则依据前面smbclient -L david的清单，我们得知可以登录暂存区这个目录(Type为Disk都表示目录)，于是执行：

```
# smbclient '\\david\暂存区'
```

登录david这台机器共享出来的暂存区(/tmp)目录。

```
Added interface ip=140.115.210.158 bcast=140.115.255.255 nmask=255.255.0.0
Server time is Sun Oct 06:03:57 1999
Timezone is UTC+8.0
```

Password: (登入Samba Server会要求你输入将要存取目录的使用者密码,例如 /tmp属于root,所以要输入root密码。若没有设密码,直接按Enter就可以进去了。成功后会出现 Samba Server的提示符号:smb:\>)

```
Domain=[SAMBA_HOUSE] OS=[UNIX] Server=[Samba 1.9.16p10]
```

smb:\>help 执行help看看指令列表。

```
ls dir lcdcd pwd
get mget putmput rename
moremask delrm mkdir
md rmdia rd prompt recurse
translate lowercase print printmode queue
qinfocancel stat quit q
exit newer archive tar blocksize
tarmode setmodehelp ? !
```

可以看出来,smbclient几乎就是ftp的变相,你可以使用dir看文件、get读取文件、put上传文件……,跟ftp没什么两样。

8. Test 8

在Linux上执行

```
# smbclient '\\hostname\sharename' -I xxx.xxx.xxx.xxx
```

分享Windows95网络邻居上的机器开放出来的文件。但必须先知道 Windows95这台机器的hostname和它开放出来的目录,以及它的IP Address,才能存取这台机器上的文件。例如:

```
# smbclient '\\chris\doom' -I 140.115.211.183
```

会连到网络邻居上主机名称为chris的机器,而连上后,用dir查看到的文件就是doom这个目录下的文件。

9. Test 9

在Windows95的机器上执行“net view \\samba_name”,其中samba_name就是你的Linux机器的hostname,如果能看到david这台机器的话,就表示你的Samba没问题了,如果发生错误的话,如“network name not found”,则必须查看nmbd有没有错误,或在Windows95网络设置的TCP/IP协议中的“wins server”加上的机器名称,或在Windows95这台机器的“C:/etc/lmhost”文件里面加上Windows95的机器网址,格式如下:

```
#C:/etc/lmhost
ip address hostname
tester
```

10. Test 10

查看能不能在Windows95的网络邻居上看到你的Samba Server,其实这是最直观的方法,只要在网络邻居上可以看到你的Server,并且可以进入你共享出来的目录,就完全没有问题了。

如果遇到不能解决的问题,无法启动Samba Server,建议你看看Samba压缩文件解压后的docs/目录里面的文件。