

Linux 网络服务

摘 要：使用 Red Hat Linux9 实现网络服务

关 键 字：代理服务器 (squid)、TCP/IP 设置、DHCP 服务器、Apache Web 服务器、FTP 服务器、PXE 无盘工作站、路由

指导老师：吴三成老师

学 生：傅 杰

1、Linux 操作系统介绍

Linux 是一种开放源代码、协作开发的类 UNIX，但同其他的 UNIX 操作系统相比又非常小。它是一种真正多用户、多任务的操作系统。该系统功能完善，可以运行在多数硬件平台上，其紧凑高效的内核能够充分发挥硬件的作用，同时对网络功能提供了广泛的支持。

Linux 最为突出的特点在于它是开放的自由软件以及其内置的网络支持。自由软件的含义是指不仅想用户提供软件本身，而且还向用户开放其全部源代码。供用户无偿使用，并允许用户对源代码作任意的修改和补充，而且不受商业机构的版权制约。Linux 使用标准的 TCP/IP 协议作为主要的网络通信协议，内建 FTP、E-mail 和 Telnet 等协议。许多 Internet 服务提供商都采用 Linux 构建 Mail Server、HTTP server 和 FTP Server 等服务器。

Linux 的主要版本包括 Red Hat、SuSe、Turbo Linux、Slackware 和 mandrake 等。其中 Red Hat Linux 是较为成熟的一种 Linux 发行版，获得了很多商业的支持，所以在硬件软件兼容上比较好，安装和使用都很方便。

在这里将以 Red Hat Linux 9 为例，来研究 Linux 的网络服务。

2、在 Linux 下安装和配置代理服务器 Squid

2.1 代理服务器 Squid 简介

代理服务器是运行在防火墙上的一种服务器程序。它是被放置在内部服务器和外部服务器之间用于转接内外主机之间的通信。

Squid 是 Linux 下最为流行的代理服务器软件，它功能强大，支持对 HTTP、FTP 和 Gopher 等协议的代理。利用 Squid 代理的 WWW 服务不仅可以过滤有害站点和限制用户对 Internet 的访问，而且 Squid 具有页面缓存功能，它接受用户的下载申请，并自动处理所下载的数据。

2.2 Squid 安装

Red Hat Linux9 自带了 Squid 软件包，按以下步骤安装 Squid:

```
#rpm -qa|grep squid      //查看是否安装了 squid
#mount /mnt/cdrom       //将第一张光盘放入光驱后挂装
#cd /mnt/cdrom/Red Hat/RPMS    //进入 RPM 目录
#rpm -ivh squid-2.3.STABLE12.i386.rm    //安装所需要的软件包
#cd;eject              //弹出光盘
```

2.3 squid 配置

Squid 有一个重要的配置文件/etc/squid/squid.conf, 该配置文件非常庞大。如果仅为中小型网络(计算机房、办公室等)提供代理服务, 并只使用一台服务器, 则只需修改以下几个主要选项:

- (1) http_port: 该选项用于定义 squid 监听 HTTP 客户请求的端口, 默认是 3128。如果使用 HTTPD 加速模式, 则应为 80。
- (2) Cache_member (bytes): 该选项用于指定 squid 可以使用的内存的理想值, 一般推荐为物理内存的 1/3。
- (3) Cache_dir Directory-Name Mbytes Level1 Level2: 该选项用于指定 squid 用于存储对象的交换空间大小以及其目录。
- (4) Directory -Name: 该选项用于指明交换空间的顶级目录, 默认为/var/spool/squid。
- (5) Mbytes: 该选项用于定义使用交换空间的总量。
- (6) Level1 Level2: 为了加快查找响应速度, 设置两级目录结构。
- (7) acl aclname acltype string1...: 该选项用于定义访问控制列表。
- (8) src: 该选项用于指明源地址, 可以是地址或地址段。
- (9) dst: 该选项用于指明目标地址。
- (10) srcdomain: 该选项用于指明客户所属的域。
- (11) dstdomain: 该选项用于指明请求服务所属的域。
- (12) time: 该选项用于指明访问时间。格式为[S|M|T|W|H|F|A] [HH1-HH2: MM2]。
- (13) port: 该选项用于指明访问端口。
- (14) proto: 该选项用于指明访问协议。
- (15) method: 该选项用于指定请求方法。
- (16) http_access: 该选项用于根据访问控制列表允许或禁止一类用户的访问。
- (17) maximum_object_size: 该选项用于设定 squid 可以接收的最大对象的大小。缺剩值为 4MB。
- (18) cache_effective_user: 该选项用于设定使用缓存的有效用户。缺省值为 nobody。

以下是一个简单的 Squid.conf 文件:

```
#squid.conf - a very basic config file for squid
#Turn logging to it's lowest level
debug_options ALL, 1
#defines a group (or Access Control list) that includes all IP addresses
acl all src 0.0.0.0/0.0.0.0
#define RAM used
cache_mem 32M
#define the cache size
cache_dir /usr/local/squid/cache 100 16 256
#allow all sites to use connect to us via HTTP
http_access allow all
#allow all sites to use us as a sibling
icp_access allow all
#test the following sites to check that we are connected
dns_testnames internic.net usc.edu cd.colorado.edu mit.edu yale.edu
#run ad the squid user
cache_effective_user squid squid
```

这个配置文件创建了 100MB 的缓存，使用了 32MB 的内存，允许所有人使用 squid，在缺省位置"/usr/local/squid/cache"缓存数据，所有缓存数据以组 squid 和用户 squid 身份保存，端口为缺省值 3128。

3、TCP/IP 网络配置

3.1 使用 redhat-config-network 配置以太网网络

Redhat-config-network 是 Red Hat Linux 9 提供的一个图形界面网络配置工具。使用该配置工具可以配置各种网络连接。单击面板上的“主菜单”-->“系统设置”-->“网络”（或者在字符界面输入 redhat-config-network），启动 redhat-config-network 管理工具。在图形界面下对网络进行配置，设置其 IP 地址，配置路由以及 DNS 客户。

将 eth0 设置为 IP: 192.168.0.254 netmask: 255.255.255.0 DNS 客户: 210.40.128.33 网关: 192.168.0.1

将 eth1 设置为 IP: 192.168.1.254 netmask: 255.255.255.0

3.2 使用命令配置以太网网络

1、ifconfig---用于配置并查看网络接口的配置情况。操作如下：

```
#ifconfig eth0 192.168.0.254 netmask 255.255.255.0 //配置 eth0 的 IP 地址，并激活
#ifconfig eth0: 1 192.168.0.150
#route add -host 192.168.0.150 dev eth0: 1 //配置 eth0 的别名设备 eth0: 1 的 IP，并添加路由。
#ifconfig eth0: 1 up //激活设备
#ifconfig eth0: 1 down //禁用设备
#ifconfig eth0 //查看设备 eth0 的配置
#ifconfig //查看所有网络设备的配置
```

2、route---用于配置并查看内核路由表的配置情况。操作如下：

```
#route add -host 192.168.1.254 dev eth0: 0
#route add -host 192.168.1.254 gw 192.168.1.254 //添加到主机的路由
#route add -net 192.168.1.254 netmask 255.255.255.248 eth0
#route add -net 192.168.1.254 netmask 255.255.255.248 gw 192.168.1.254
#route add -net 192.168.1.0/24 eth1 //添加到网络的路由
#route add default gw 192.168.1.254 //添加默认网关
#route //查看内核路由表的配置
#route del -host 192.168.1.254 dev eth0: 0
#route del -host 192.168.1.254 gw 192.168.1254 //删除路由
```

3、traceroute---用于显示数据包到达目的主机所经过的路由。

```
#traceroute www.163.com
```

4、ping---用于测试网络的连通性。

```
#ping www.163.com
```

5、netstat---用于显示网络状态信息。

6、hostname---用于更改主机名。

```
#hostname cactus
```

7、arp---用于配置并查看 ARP 缓存。

4、DHCP 服务器

4.1 DHCP 概述

在 TCP/IP 网络上，每台工作站要在存取网络上的资源之前，都必须进行基本的网络主要参数配置，如 IP 地址、子网掩码、缺省网关、DNS 客户等，还可能是一些附加的信息，如 IP 管理策略之类。而这些对于一个较大型的网络而言，网络管理和维护的任务是很繁重的。

动态主机配置协议 DHCP 则是为了实现 IP 的自动配置而设计的协议。他可以为客户机自动分配 IP 地址、子网掩码、缺省网关、DNS 等 TCP/IP 参数。目的就是为了减轻 TCP/IP 网络规划、管理和维护的负担，解决 IP 地址空间缺乏的问题。

4.2 DHCP 服务器的安装

```
#rpm -qa|grep dhcp //查看是否安装了 dhcp
#mount /mnt/cdrom //将第 2 张盘放入光驱后挂装
#cd /mnt/cdrom/red hat/rpms //进入 rpms 目录
#rpm -ivh dhcp-3.0p11-23.i386.rpm //安装所需要的 rpm 包
#cd;eject
```

4.3 DHCP 配置

DHCP 服务器的配置核心文件是/etc/dhcpd.conf，每次启动 DHCP 服务器都要读取该文件，在 dhcpd.conf 中对 DHCP 服务器做了很多定义。

以下是 DHCP 配置文件的实例：

```
#sample /etc/dhcpd.conf;
default-lease-time 1200; //默认 IP 地址出租时间为 1200s
max-lease-time 9200; //最大 IP 地址出租时间为 9200s（若是出租时间永不过期则为-1）
option subnet-mask 255.255.255.0; //子网掩码地址
option broadcast-address 192.168.1.255; //广播地址
option routers 192.168.1.254; //路由 IP 地址
option domain-name-servers 192.168.1.254; //DNS 的 IP 地址
option domain-name "mydomain.org"; //域名选项
subnet 192.168.1.0 netmask 255.255.255.0{
range 192.168.1.10 192.168.1.100;
range 192.168.1.150 192.168.1.200;
} //子网掩码为 255.255.255.0，子网为 192.168.1.0 的网络，{ }内部是分配的 IP 地址。
```

该文件定义了服务器发送给 DHCP 客户机以下参数：用 255.255.255.0 作为子网掩码，用 192.168.1.255 作为广播地址，用 192.168.0.1 作为默认网关，用 192.168.1.254 做为 DNS 服务器。

有时需要在一台安装有两张网卡的主机上安装 DHCP 服务，由于对外的网卡不需要提供 DHCP 服务，则设置如下：

```
subnet 192.168.1.0 netmask 255.255.255.0{
range 192.168.1.254;
default-lease-time 86400;
max-lease-time 259200;
option subnet-mask 255.255.255.0;
```

```

option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.254;
}
subnet 202.102.34.102 netmask 255.255.255.255{
}

```

5、Apache Web 服务器

5.1 Apache 的安装

Red Hat Linux 9 自带了 Apache2.0，以下是 Apache 的安装步骤：

```

#rpm -qa|grep httpd
#mount /mnt/cdrom //将第 1 张光盘放入光驱后挂装
#cd /mnt/cdrom/red hat/rpms
#rpm -ivh httpd-2.0.40.21.i386.rpm
#rpm -ivh httpd-manual-2.0.40-21.i386.rpm
#cd;eject

```

在安装时，Apache 采用了一系列的缺省值，系统启动后，WWW 服务器已经运行。将装上 linux+apache 的主机联入 internet 后，把自己的主页存到/home/httpd 目录下即可。

5.2 Apache+PHP

安装 php

```

#mount /mnt/cdrom //将第 1 张光盘放入光驱后挂装
#cd /mnt/cdrom/red hat/rpms
#rpm -ivh curl-7.9.8-5.i386.rpm
#rpm -ivh gd-1.8.4-11.i386.rpm //安装 php 所需的 curl 和 gd
#rpm -ivh php-4.2.2-17.i386.rpm //安装 php
#rpm -ivh php-imap-4.2.2-17.i386.rpm //安装 php 的 imap 支持包
#cd;eject

#mount /mnt/cdrom //将第 2 张光盘放入光驱后挂装
#cd /mnt/cdrom/red hat/rpms
#rpm -ivh php-manual-4.2.2-17.i386.rpm //安装 php 手册
#rpm -ivh php-mysql-4.2.2-17.i386.rpm //安装 php 的 mysql 支持包
#rpm -ivh php-pgsql-4.2.2-17.i386.rpm //安装 php 的 pgsq 支持包
#cd;eject

```

在文件/usr/local/apache/conf/httpd.conf 中<IfModule mod_mime.c>中添加以下语句：

```

addtype application/x-httpd-php .php
addtype application/x-httpd-php-source .phps

```

然后修改 php 配置文件 php.ini

```
register_globals=on
```

重启 apache 服务器

```
#httpd restart
```

然后写个 php 测试页 info.php 内容如下：

```
<?php
  phpinfo ( )
?>
```

测试 php: 打开浏览器, 在地址栏上输入 192.168.0.254/info.php
如果能看到 php 的信息, 则说明 apache+php 安装成功。

5.3 Apache+jsp

整合 JDK 和 TOMCAT 环境

环境: Red Hat Linux 9 apache 2.0 php4

需要软件: (在/usr/local 下安装) apache 安装路径为/usr/local/apache

1. 安装 jdk 1.4.2

```
#cd /usr/local/
#wget ftp: //202.96.64.158/pub/j2sdk-1_4_2_03-linux-i586.bin
#chmod a+x j2sdk-1_4_2_03-linux-i586.bin
#./j2sdk-1_4_2_03-linux-i586.bin
```

将所下载的 j2sdk 复制到目录/usr/local/下面以/j2sdk 为目录

2. 安装 tomcat

```
#cd /usr/local/
#wget http: //apache.linuxforum.net/dist/jakarta/tomcat-4/v4.1.29/bin/
jakarta-tomcat-4.1.29.tar.gz
#tar zxf jakarta-tomcat-4.1.29.tar.gz
```

将下载的 tomcat 解压后复制到/usr/local/下以/tomcat 为目录

3. 为 jdk 和 tomcat 建立链接

```
ln -s j2sdk jdk
ln -s tomcat tomcat
```

4. 设置环境变量

vi /etc/profile 在最后加入以下内容, 并在系统中运行一下

```
PATH=$PATH: /usr/local/j2sdk/bin: /usr/local/j2sdk/jre/bin
JAVA_HOME=/usr/local/j2sdk
export JAVA_HOME
CLASSPATH="./: /usr/local/j2sdk/lib: /usr/local/j2sdk/jre/lib"
export CLASSPATH
CATALINA_HOME=/usr/local/tomcat
export CATALINA_HOME
```

5. 编译安装 Connector

```
#cd /usr/local
#wget http: //apache.linuxforum.net/dist/jakarta/tomcat-4/v4.1.29/src/
jakarta-tomcat-connectors-4.1.29-src.tar.gz
#tar zxf jakarta-tomcat-connectors-4.1.29-src.tar.gz
```

```
#cd jakarta-tomcat-connectors-4.1.29-src/webapp // 进入指定的目录
#wget http://apache.linuxforum.net/dist/apr/apr-0.9.4.tar.gz
#tar zcf apr-0.9.4.tar.gz
#mv apr-0.9.4.apr
./support/buildconf.sh // 生成编译配置文件
./configure --with-apxs=/usr/local/apache/bin/apxs --with-tomcat=/usr/local/tomcat // 生成
Makefile 编译文件 make 编译生成 mod_webapp.so (mod_webapp.so 的生成和 apache 版本
相关, 到相应目录下找)
cp apache-2.0/mod_webapp.so /usr/local/apache/modules
并检查文件/usr/local/apache/modules/mod_webapp.so 的属性, 应与其它 Modules 文件属性一
致, 如果不相同, 则使用 chmod chgrp chown 来修改
```

6. 独立环境的测试

(1) 测试 Java 的运行状况, 如下命令:

```
java -version
```

看到版本号则表示 JDK 安装成功

(2) 测试 Jakarta-Tomcat 的运行状况, 如下命令:

```
$CATALINA_HOME/bin/startup.sh // 启动 tomcat 服务
```

使用任意一浏览器, 输入服务器地址和端口号 8080 后回车, 应该可以看到 tomcat 的默认主页 <http://192.168.0.254:8080>

```
$CATALINA_HOME/bin/shutdown.sh // 关闭 tomcat 服务
```

7. 整合安装设置

修改 Apache 的配置文件

(1) 修改/etc/httpd/conf/httpd.conf 文件

```
vi /usr/local/apache/conf/httpd.conf
```

在# Dynamic Shared Object (DSO) Support 之后, 添加如下内容

```
LoadModule webapp_module libexec/mod_webapp.so
```

```
#LoadModule webapp_module modules/mod_webapp.so (apache2.0 用)
```

(2) 测试配置和 Modules, 执行以下命令

```
apachectl configtest
```

如果出现 Syntax OK, 则证明 Module 安装和配置成功了!

3) 修改/etc/httpd/conf/httpd.conf 文件

在文件尾部添加如下内容

```
ServerName localhost: 8008
```

```
WebAppConnection warpConnection warp localhost: 8008
```

```
WebAppDeploy examples warpConnection /examples
```

4) 再次测试配置, 执行以下命令

```
apachectl configtest
```

如果出现 Syntax OK, 则证明安装成功!

5) 修改 Tomcat 的配置文件

```
cd $CATALINA_HOME/conf/
```

```
mv server.xml server.xml.bak
```

```
cp server-noexamples.xml.config server.xml //server-noexamples.xml 里是标准默认配置。
```

重新启动所有服务

```
$CATALINA_HOME/bin/shutdown.sh
```

```
$CATALINA_HOME/bin/startup.sh
```

```
/etc/init.d/httpd restart
```

http: //192.168.0.254/examples/jsp/index.html, 如果能够显示正常的网页, 则安装成功

在浏览器中输入如下地址 http: //服务器 IP/examples/jsp/jsptoserv/jspotoservlet.jsp 如果能够显示正常的网页和时间, 则 jsp 和 Servlet 安装成功

6、FTP 服务器

6.1 FTP 协议 FTP 服务器简介

FTP 协议定义了一个在远程计算机系统和本地计算机系统之间传输文件的一个标准。FTP 运行在 OSI 模型的应用层, 并利用传输控制协议 TCP 在不同的主机之间提供可靠的数据传输。

在使用 FTP 服务时, 要传输文件的用户需要登陆 FTP 站点后方能访问远程服务器上的文件。

6.2 配置 vsftpd

RedHat 自带了 vsftpd 的 ftp 系统。下面就简要的说一下它的配置方法。

相关配置文件/etc 目录下的 vsftpd.conf, vsftpd.ftpuser, vsftpd.user_list, /etc/xinetd.d/vsftpd

配置 vsftpd 主要是对主配置文件/etc/vsftpd.conf 的修改, 相关参数如下:

```
anonymous_enable=YES      //是否允许匿名 ftp, 如否则选择 NO
local_enable=YES          //是否允许本地用户登录
local_umask=022          //默认的 umask 码
anon_upload_enable=YES    //是否允许匿名 ftp 用户访问
anon_upload_enable=YES    //是否允许匿名上传文件
anon_mkdir_write_enable=YES //是否允许匿名用户有创建目录的权利
dirmessage_enable=YES    //是否显示目录说明文件, 默认是 YES 但需要手工创建.message 文件
xferlog_enable=YES       //是否记录 ftp 传输过程
connect_from_port_20=YES //是否确信端口传输来自 20 (ftp-data)
chown_upload=YES
chown_username=username  //是否改变上传文件的属主, 如果是需要输入一个系统用户名, 你可以把上传的文件都改成 root 属主
xferlog_file=/var/log/vsftpd.log // ftp 传输日志的路径和名字默认是/var/log/vsftpd.log
xferlog_std_format=YES   //是否使用标准的 ftp xferlog 模式
idle_session_timeout=600 //设置默认的断开不活跃 session 的时间
data_connection_timeout=120 //设置数据传输超时时间
nopriv_user=ftpsecure    //运行 vsftpd 需要的非特权系统用户默认是 nobody
async_abor_enable=YES    //是否允许运行特殊的 ftp 命令
ascii_upload_enable=YES
ascii_download_enable=YES //是否使用 ascii 码方式上传和下载文件
ftpd_banner=Welcome to chenlf FTP service. //定制欢迎信息
deny_email_enable=YES
banned_email_file=/etc/vsftpd.banned_emails //是否允许禁止匿名用户使用某些邮件
```

地址，如果是输入禁止的邮件地址的路径和文件名

```
chroot_list_enable=YES
```

chroot_list_file=/etc/vsftpd.chroot_list //是否将系统用户阻止在自己的 home 目录下，
如果选择了 yes 那么

```
chroot_list_file=/etc/vsftpd.chroot_list 中列出的是不 chroot 的用户的列表
```

max_clients=Number //如果以 standalone 模式起动，那么只有\$Number 个用户可以连接，
其他的用户将得到错误信息，默认是 0 不限止

```
message_file //设置访问一个目录时获得的目录信息文件的文件名，默认是.message
```

以下是配置文件 vsftpd.conf 的实例.允许匿名 FTP，允许上传，chroot 用户目录，上传
文件属主改为 root

```
anonymous_enable=YES
```

```
local_enable=YES
```

```
write_enable=YES
```

```
local_umask=022
```

```
anon_upload_enable=YES
```

```
anon_mkdir_write_enable=YES
```

```
dirmmessage_enable=YES
```

```
xferlog_enable=YES
```

```
connect_from_port_20=YES
```

```
chown_uploads=YES
```

```
chown_username=root
```

```
xferlog_file=/var/log/vsftpd.log
```

```
xferlog_std_format=YES
```

```
pam_service_name=vsftpd
```

```
chroot_local_user=YES
```

7、pxe Linux 无盘工作站架设实例

架设 Linux 无盘工作站实例，服务器和客户端都使用 Linux。主要涉及：DHCP TFTP
SYSLINUX NFS 客户端是支持 PXE 方式启动的刀片，用 Linux 作为服务器，服务器配置如
下：

1、安装 DHCP 服务器 dhcpd

2、配置/etc/dhcpd.conf 文件，下面是我机器上的文件：

```
max-lease-time -1;
```

```
default-lease-time -1; //IP 地址永不过期
```

```
option subnet-mask 255.255.255.0;
```

```
option routers 192.168.1.254;
```

```
option domain-name-servers 192.168.1.254;
```

```
# option netbios-name-servers 10.10.10.10;
```

```
option broadcast-address 255.255.255.255;
```

```
# option dhcp-class-identifier "PXEClient";
```

```
# option vendor-encapsulated-options 01: 04: 00: 00: 00: 00: ff;
```

```
# option option-135 "start";
```

```
subnet 192.168.1.0 netmask 255.255.255.0{
```

```

range 192.168.1.10 192.168.1.100;
host blade01 {
    hardware ethernet 00: e0: 43: 00: 02: 00;
    fixed-address 192.168.1.11;
    filename "/tftpboot/pxelinux.0";           //启动映象文件
}
host blade02 {
    hardware ethernet 00: e0: 43: 00: 02: 02;
    fixed-address 192.168.1.12;
    filename "/tftpboot/pxelinux.0";
}
host blade03 {
    hardware ethernet 00: e0: 43: 00: 02: 04;
    fixed-address 192.168.1.13;
    filename "/tftpboot/pxelinux.0";
}

```

说明: dhcp 客户得到 IP 地址后用 TFTP 协议从服务器上下载启动映象文件。我用 syslinux 工具包里边的 pxelinux 来作为远程启动的 loader。

3、配置 tftp server

使用 pxelinux 作引导工具需要支持 TSIZE 参数的 tftp server。
通过 xinetd 来使用 tftp 服务, 我的/etc/xinetd.conf 文件如下:

```

.....
service tftp
{
    socket_type      = dgram
    protocol        = udp
    wait            = yes
    user            = root
    server          = /usr/sbin/in.tftpd
}

```

4、配置 PXELINUX

先安装 syslinux 软件包。

将 pxelinux.0 拷贝到/tftpboot/目录下, 然后建立/tftpboot/syslinux.cfg/目录。该目录下存放配置文件。pxelinux 使用 ip 地址的十六进制表示来作为该 ip 地址的配置文件的文件名。如 blade01 的 ip 地址为 192.168.0.11, 配置文件名为 C0A5000B, 内容为:

```

default linux
label linux
    kernel vmlinuz
    append ip=dhcp root=/dev/nfsroot nfsroot=192.168.0.254: /remote/blade01 vga=normal

```

5、配置 nfs

为每个刀片建立一个根目录, 在该刀片的 pxelinux 配置文件里指定了从这个 nfs export 的目录启动。

该根目录里应把标准的目录都建好, 另外需要重新 mount 的 usr, home, public 等目录也要 export。

我的 /etc/exports 文件:

```
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports (3) .
/remote/blade01 blade01 (rw, async, no_root_squash)
/remote/blade02 blade02 (rw, async, no_root_squash)
/remote/blade03 blade03 (rw, async, no_root_squash)
/remote/root * (rw, async, no_root_squash)
/remote/home * (rw, async, no_root_squash)
/usr * (ro, async, no_root_squash)
/sbin * (ro, async, no_root_squash)
/bin * (ro, async, no_root_squash)
/lib * (ro, async, no_root_squash)
/home * (ro, async, no_root_squash)
```

- 6、为每个刀片修改它的/etc/fstab 文件，以 blade01 为例，它的 nfs root 是/remote/blade0 1
/remote/blade01/etc/fstab 文件如下:

```
# /etc/fstab: static file system information.
#
192.168.1.254: /remote/blade01 / nfs defaults, intr 0 1
192.168.1.254: /remote/root /root nfs defaults, intr 0 1
192.168.1.254: /remote/home /home nfs defaults, intr 0 1
192.168.1.254: /bin /bin nfs defaults, intr 0 1
192.168.1.254: /usr /usr nfs defaults, intr 0 1
192.168.1.254: /sbin /sbin nfs defaults, intr 0 1
192.168.1.254: /lib /lib nfs defaults, intr 0 1
none /proc proc defaults, intr 0 1
```

同时还要为每个刀片修改它的网络配置文件，配置 ip 地址，启动两块网卡等等。

7、编译内核

刀片用的内核，应该支持 Kernel Level Auto Configuration 的 DHCP 协议，支持 NFS，支持 NFS ROOT，假设编译好的内核为 vmlinuz，将它拷贝到/tftpboot/目录下

8、基于 Linux 的路由器

8.1 路由器简介

Internet 是通过路由器互联的网络和主机组成的一个庞大的集合，路由器 (router) 是连接两个或更多包 (分组) 交换网络的专用计算机。当从一个网络想另一个网络传输发信息流时，路由器起到一个包交换媒介的作用。

8.2 Linux 环境路由器配置实例 (静态路由的配置实例)

1. 条件和要求

两个以太网网卡所对应的 IP 地址分别是 192.168.0.0/24 和 192.168.1.0/24，子网掩码是 255.255.255.0，现在要求配置一台路由器，实现这两个局域网的互访。

2. 配置步骤:

(1) 确定一台将要充当路由器的计算机，安上两张网卡后，再安装 linux 操作系统，在安装过程中，安装系统会自动识别着两张网卡。

(2) 配置网卡参数：将两张网卡的 IP 地址分别设置为 192.168.0.254 和 192.168.1.254，

分别属于两个局域网。在管理员状态下，执行以下命令：

```
#ifconfig eth0 192.168.0.254 broadcast 192.168.0.255 netmask 255.255.255.0
```

```
#ifconfig eth0 192.168.1.254 broadcast 192.168.1.255 netmask 255.255.255.0
```

(3) 添加路由：

```
#route add -net 192.168.0.0 netmask 255.255.255.0 dev eth0
```

```
#route add -net 192.168.1.0 netmask 255.255.255.0 dev eth1
```

(4) 设置网关：

将 192.168.0.0/24 的网络内部所有主机的网关设为 192.168.0.254，将 192.168.1.0/24 的网络内部所有主机的网关设为 192.168.1.254。

参考文献

(1) SmarTraining 工作室、梁如军、丛日权等编著《Red Hat Linux 9 网络服务》，机械工业出版社，2004/1。

(2) 黄继海、杨凯、左军、张茜萍等编著《国家网络技术水平认证考试指南--Linux 篇》，河南科学技术出版社，2003/9。

(3) <http://www-900.ibm.com/developerWorks/cn/linux/l-tip-prompt/l-pex/index.shtml>

(4) <http://www.knowsky.com/index.asp>