



中标软件

中标麒麟高级服务器操作系统 V6.0

网络服务用户手册

中标软件有限公司

2014 年 04 月 20 日

目录

1 WEB 服务器配置和使用	1
1.1 Apache (HTTP) 配置	1
1.1.1 基本配置	2
1.1.2 虚拟主机的缺省默认设置	3
1.1.3 虚拟主机设置	12
1.1.4 服务器设置	19
1.1.5 调整性能	21
1.1.6 保存设置	23
1.1.7 其它资源	23
1.2 Apache 命令和模块	24
1.2.1 启动和终止 httpd	24
1.2.2 httpd.conf 中的配置命令	25
1.2.3 缺省模块	43
1.2.4 添加模块到服务器	44
1.2.5 使用虚拟主机	45
1.2.6 附加资源	47
2 DNS 配置和使用	49
2.1 管理 DNS 设置	49
2.2 BIND 配置	51
2.3 Named.conf 配置文件	51
2.4 配置示例	53
2.4.1 主配置文件	53
2.4.2 域名解析	56
2.4.3 IP 地址逆向解析	56
2.5 图形化 DNS 服务器配置工具	57
3 SAMBA 配置和使用	61
3.1 配置 Samba 服务器	61
3.1.1 图形化配置	61
3.1.2 加密口令	66
3.1.3 启动和停止服务器	68
3.2 挂载共享	69

3.3 手动配置 Samba.....	69
3.4 连接到 Samba 共享.....	70
3.5 如何在中标麒麟高级服务器操作系统下使用加密口令配置 Samba.....	70
3.6 附加资源.....	71
4 NFS 配置和使用	73
4.1 为什么使用 NFS?	73
4.2 安装 NFS 文件系统	73
4.2.1 使用/etc/fstab 安装 NFS 文件系统	73
4.2.2 使用 autofs 安装 NFS 文件系统	74
4.2.3 ACL 支持.....	76
4.3 导出 NFS 文件系统	76
4.3.1 图形化配置工具.....	76
4.3.2 命令行配置.....	80
4.3.3 主机名格式.....	81
4.3.4 启动和停止服务器.....	81
4.4 附加资源.....	82
5 邮件服务器配置和使用.....	83
5.1 电子邮件协议.....	83
5.1.1 邮件传输协议.....	83
5.1.2 邮件存取协议.....	84
5.2 电子邮件程序分类.....	87
5.2.1 邮件传输代理.....	87
5.2.2 邮件投递代理.....	88
5.2.3 电子邮件用户代理.....	88
5.3 邮件传输代理.....	88
5.3.1 Postfix	88
5.3.2 Sendmail	91
5.3.3 邮件工具 (Fetchmail)	97
5.3.4 邮件传输代理 (MTA) 配置	102
5.4 邮件投递代理.....	102
5.4.1 Procmail 配置	103
5.4.2 Procmail Recipes	104
5.5 邮件用户代理.....	111

5.5.1 安全通信.....	111
5.6 更多信息资源.....	113
5.6.1 已安装文件.....	113
5.6.2 有用网址.....	114
5.6.3 相关书籍.....	114
6 VSFTPD 配置和使用.....	116
6.1 关于 vsftpd.....	116
6.2 Vsftpd 的常用配置文件.....	117
6.3 启动和中止 vsftpd.....	117
6.4 启用 vsftpd 的多个副本.....	118
6.5 vsftpd 配置选项.....	119
6.5.1 程序选项.....	119
6.5.2 登录选项和访问控制.....	120
6.5.3 匿名用户选项.....	121
6.5.4 本地用户选项.....	122
6.5.5 目录选项.....	123
6.5.6 文件传输选项.....	124
6.5.7 日志选项.....	124
6.5.8 网络选项.....	125
6.6 其它资源.....	127
6.6.1 安装的文档.....	127
6.6.2 有用的网址.....	128
7 SQUID 配置和使用.....	129
7.1 Squid 简介.....	129
7.2 Squid 的编译和运行.....	129
7.3 squid.conf 文件的配置.....	130
7.4 运行 Squid.....	131
8 LDAP 配置和使用.....	132
8.1 什么是 LDAP?.....	132
8.2 LDAP 的优点.....	135
8.2.1 OpenLDAP 2.0 的增强特性.....	135
8.3 LDAP 术语.....	135
8.4 OpenLDAP 守护进程和应用程序.....	136

8.4.1 NSS、PAM 和 LDAP	138
8.4.2 PHP4、LDAP 和 Apache HTTP 服务器.....	138
8.4.3 LDAP 客户端应用	139
8.5 OpenLDAP 配置文件	139
8.6 etc/openldap/schema/目录.....	140
8.7 建立 OpenLDAP	140
8.7.1 编辑/etc/openldap/slapd.conf	141
8.8 使用 OpenLDAP 配置认证系统	142
8.8.1 迁移老的认证信息到 LDAP 格式	143
8.9 从早期版本迁移目录.....	144
8.9.1 迁移 1.x 目录.....	145
8.10 附加资源.....	145
8.10.1 已安装的文档.....	145
8.10.2 有用的网址.....	146
8.10.3 相关的书.....	146
9 SMTP 配置和 IMAP 配置	147
9.1 SMTP 介绍	147
9.2 IMAP 介绍.....	147
9.3 SMTP 配置和 IMAP 配置	147
10 NTP	149
10.1 在网络上同步日期和时间.....	149
11 TFTP	151
11.1 安装 TFTP 服务	151
11.2 TFTP 服务的启动与停止	151
11.2.1 tftp 服务运行	151
11.2.2 tftp 服务停止	152
11.3 使用 TFTP 服务上传和下载文件	153
12 SNMP.....	154
12.1 snmp 服务的启动与停止.....	154
13 SSH	155
13.1 SSH 协议	155
13.1.1 为什么使用 SSH?	155
13.1.2 主要功能.....	156

13.1.3 协议版本.....	157
13.1.4 SSH 连接的事件序列	157
13.2 OpenSSH 配置.....	159
13.2.1 配置文件.....	159
13.2.2 启动 OpenSSH 服务器.....	160
13.2.3 请求 SSH 远程连接	161
13.2.4 使用基于密钥的身份认证.....	161
13.3 OpenSSH 客户端.....	167
13.3.1 使用 SSH 功能	167
13.3.2 使用 scp 实用工具	168
13.3.3 用 sftp 实用工具.....	169
13.4 不只是安全命令行.....	170
13.4.1 X11 转发.....	170
13.4.2 端口转发.....	171
13.5 额外信息资源.....	172
13.5.1 已安装文件.....	172
13.5.2 有用的网站.....	173
14 DHCP 配置.....	174
14.1 为什么要用 DHCP?	174
14.2 DHCP 服务器的配置.....	174
14.2.1 配置文件.....	175
14.2.2 租赁数据库.....	178
14.2.3 启动和终止服务器.....	179
14.2.4 DHCP 中继代理.....	180
14.3 配置 DHCP 客户端.....	180
14.4 多地址 DHCP 服务器的配置.....	181
14.4.1 主机配置.....	183
14.5 DHCPforIPv6(DHCPv6)	185
14.6 更多信息资源.....	185
14.6.1 已安装的文件.....	185
15 MYSQL 配置和使用.....	187
15.1 MYSQL 的主要特性.....	187
15.2 数据库管理.....	190

15.2.1 MySQL 服务器和服务端启动脚本.....	190
15.2.2 服务器端脚本和实用工具概述.....	190
15.2.3 MYSQLD-MAX 扩展 MySQL 服务器	192
15.2.4 MYSQLD_SAFE: MySQL 服务器启动脚本.....	195
15.3 MySQL 程序概述.....	199
15.3.1 MySQL 程序概述.....	199
15.3.2 调用 MySQL 程序.....	200
15.3.3 指定程序选项.....	201
16 POSTGRESQL 配置和使用	210
16.1 基本体系概念.....	210
16.2 创建一个数据库.....	210
16.3 访问数据库.....	212
16.4 POSTGRESQL 的简单使用	213
16.4.1 基础概念.....	213
16.4.2 创建新表.....	214
16.4.3 向表中添加行.....	215
16.4.4 查询一个表.....	216
16.4.5 在表之间连接.....	216
16.4.6 聚集函数.....	218
16.4.7 更新.....	220

1 WEB 服务器配置和使用

Apache HTTP 服务器是由 Apache Software Foundation 开发的一个商业级开放源码式 Web 服务器(<http://www.apache.org/>)。中标麒麟高级服务器操作系统包含了 Apache HTTP Server 2.2。

Apache HTTP 服务器默认安装的配置文件在大多数情况下不需要任何改变即可应用。以下简要介绍在配置文件（/etc/httpd/conf/httpd.conf）中使用的很多命令。

 注意：如果您使用图形界面的 system-config-httpd，请不要在 HTTP 图形配置工具重新生成文件时再手动编辑 Apache HTTP 服务器的配置文件/etc/httpd/conf/httpd.conf。

1.1 Apache (HTTP) 配置

Apache 图形配置工具要求 root 用户权限，使用以下任意一种方法均可打开该 HTTP 配置工具：

- 1) 在桌面上，点击【启动】→【系统】→【管理】→【服务器设置】→【HTTP 设置工具】。
- 2) 在 shell 提示中输入指令 system-config-httpd（即 GNOME 终端）。

 注意：不要修改 httpd.conf 文件：如果需要使用 httpd 的图形配置工具，不要再手工修改/etc/httpd/conf/httpd.conf Apache 配置文件。Apache 配置工具会在保存和退出后自动生成该配置文件。

Apache 配置工具允许的配置文件 httpd.conf 被存于/etc/httpd/conf 而不是老的 srm.conf 或 access.conf。您可以使用它来配置您的 Apache 网络服务器。通过图形界面，您可以对诸如虚拟主机、登录属性、最大联线数等 Apache 指令进行配置。

Apache 配置工具只能对部分（与中标麒麟服务器操作系统同时发行的）模块进行配置。该工具将无法配置任何另行添加或安装的模块。

一般来说，使用 Apache 配置工具对 Apache 网络服务器进行配置包括以下步骤：

- 1) 在【主要设置】标签下对基本设置进行配置。

2) 在【**虚拟主机**】标签下，对缺省虚拟主机进行配置。

如果您希望提供超过一个的 URL 或者虚拟主机，选择添加其他的虚拟主机。

3) 在【**服务器**】标签下，对服务器设置进行配置。

4) 在【**调整性能**】下，对连接设置进行配置。

将所有必要的文件剪贴到文档根目录和 cgi-bin 目录中，并保存设置。

1.1.1 基本配置

使用【**主要设置**】选项卡以配置服务器基本设置，如图 1-1 所示。

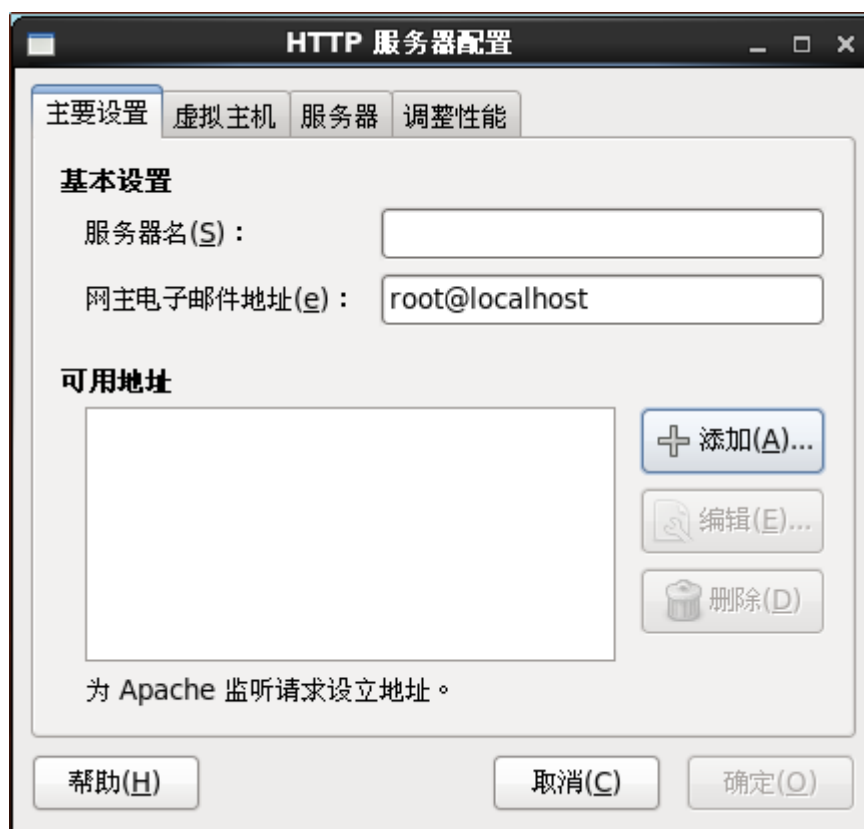


图 1-1 HTTP 服务器配置主要设置

在【**服务器名**】栏中输入一个您有权使用的完整域名。该选项对应于 httpd.conf 文件中的 ServerName 指令。ServerName 指令将在生成重定向 URL 的时候被用到，并设定网络服务器的主机名。如果您没有输入服务器名称，Apache 将使用系统的 IP 地址。设定的服务器名称并不一定要等同于服务器所在 IP 地址所对应的域名。例如，即使您的服务器真正的 DNS 名是 foo.your_domain.com，您仍然可以将服务器名称设置为 www.you_domain.com。

在【**网主电子邮件地址**】栏下输入服务器维护人员的邮件地址。该选项对应

于 httpd.conf 文件中的 ServerAdmin 指令。如果您在服务器的错误页面中设置了 email 地址，则用户将使用该地址向管理人员汇报错误信息。该地址的缺省设置是 root@localhost。

使用【可用地址】栏以定义 Apache 用于接收要求的端口。该选项对应于 httpd.conf 文件中的 Listen 指令。对于非安全网络通讯，Apache 的缺省设置为端口 80 和 8080。点击【添加】按钮以定义新的将接收要求的端口，一个如图 1-2 显示的窗口将出现。选择【监听所有地址】选项以监听在定义的端口上所有的 IP 地址，或在【地址】栏中指定一个单一的 IP 地址，服务器将只接受来自该 IP 地址的连接。请指定一个 IP 地址并对应于某个端口号。如果您需要使某个端口号对应多个 IP 地址，请为每个 IP 地址创建一个条目。在可能的情况下，尽量使用 IP 地址，而不是域名以防止 DNS 寻找失败的可能性。如果在 DNS 和 Apache 方面需要更多的资料，可参考 <http://httpd.apache.org/docs/dns-caveats.html>。地址栏中输入星号 (*) 将等价于选择监听所有地址。点击【编辑】按钮后将弹出一个与点击添加按钮非常类似的窗口。若要删除某条目，只需选中该条目后点击【删除】按钮即可。

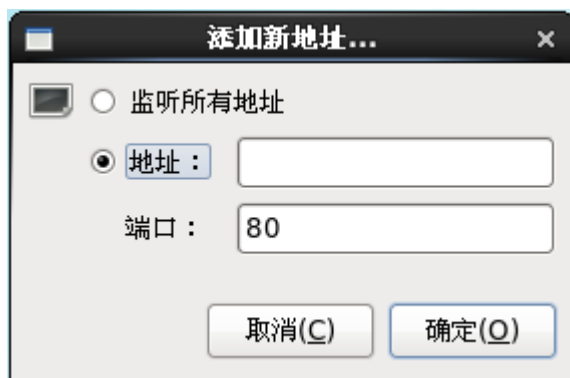


图 1-2 添加新地址

 注意：如果您的设置要求 Apache 监听一个小于 1024 的端口，则您必须作为 root 用户才能将其启动。而对于大于或等于 1024 的端口，可以作为一个普通用户启动 httpd。

1.1.2 虚拟主机的缺省默认设置

在输入服务器名称、管理人员 email、有效地址以后，点击【虚拟主机】标签并点击【编辑默认设置】按钮，如图 1-3 的窗口将会弹出。在此窗口下设置网络服务器的缺省设置。若需要添加一个虚拟主机，则您对该虚拟主机配置的设置

置将拥有优先。如果有任何指令在虚拟主机设置中没有被定义，则使用缺省设置值。

1.1.2.1 网页选项

目录页搜寻列表和错误页码的缺省数值可以适应绝大多数的服务器。如果您对这些设置没有把握，请不要对其进行修改。

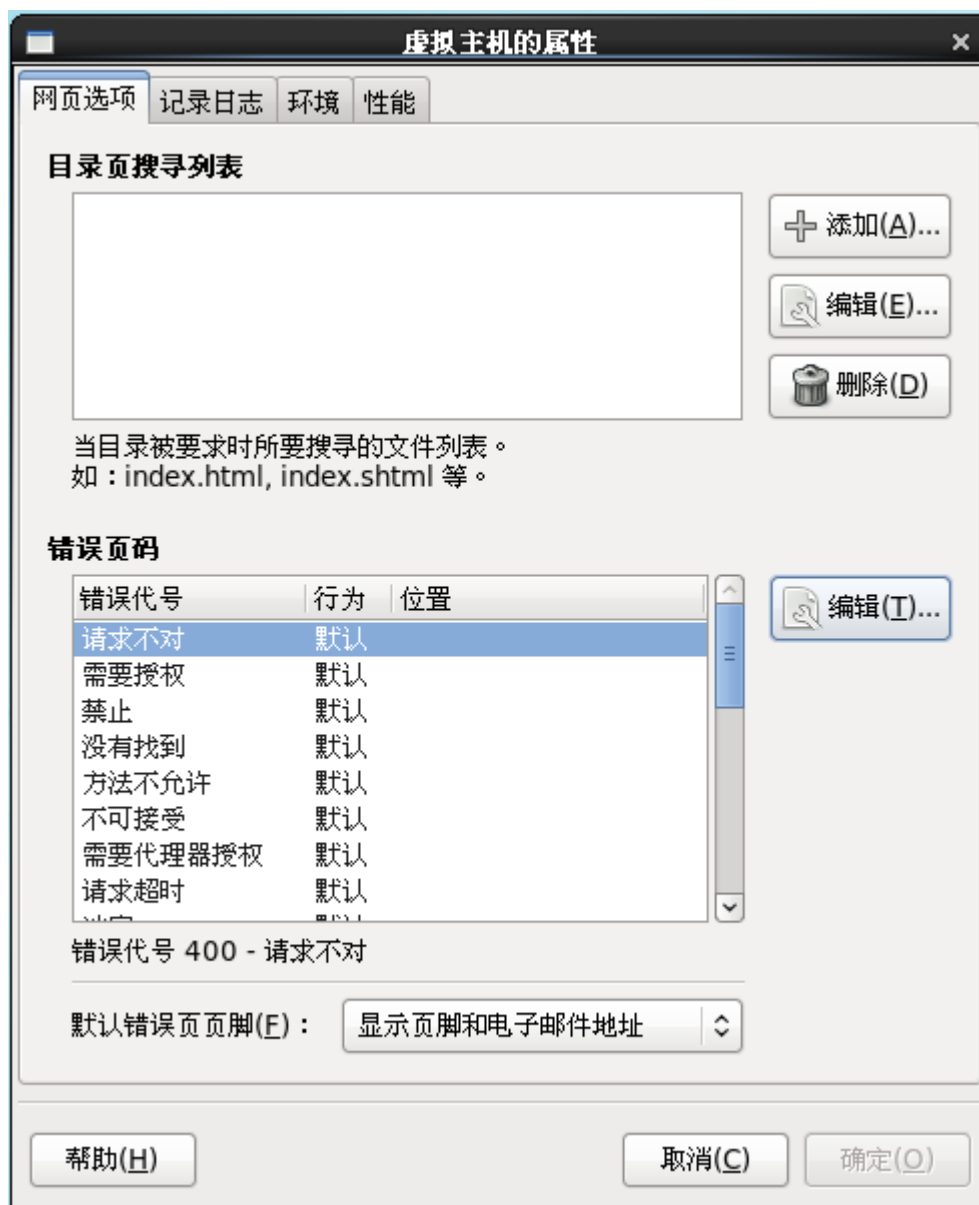


图 1-3 虚拟主机的设置

【目录页搜索列表】中所列的条目定义了 DirectoryIndex 指令的内容。当用户在目录名后输入斜划线 (/) 以寻找该目录的索引时，服务器提供的索引页面将以 DirectoryIndex 为缺省设置。

例如：当用户输入以下地址 `http://your_domain/this_directory/`，如果 `DirectoryIndex` 页面存在，服务器将提供该页面；否则服务器将自动生成一个目录列表。服务器将会试着寻找在 `DirectoryIndex` 指令中被列出的文件，并缺省显示第一个找到的符合要求的文件。如果服务器未能找到任何符合要求的文件，而同时该目录的 `Options Indexes` 已被设置，则该服务器将自动生成一个包含了所有该目录下的文件及子目录列表的 HTML 文件。

使用【**错误页码**】部分的配置可以使 Apache 在出现错误的情况下，将用户重定向到另一个本地/远程的 URL 去。这个选项对应于 `ErrorDocument` 指令的内容。如果在用户尝试连接 Apache 网络服务器的时候出现了错误，则根据缺省设置，保存在【**错误页码**】部分的一条简短错误信息将被显示给用户。如果要覆盖以上的缺省设置，请选择错误编码并点击【**编辑**】按钮，将弹出如图 1-4 的窗口：

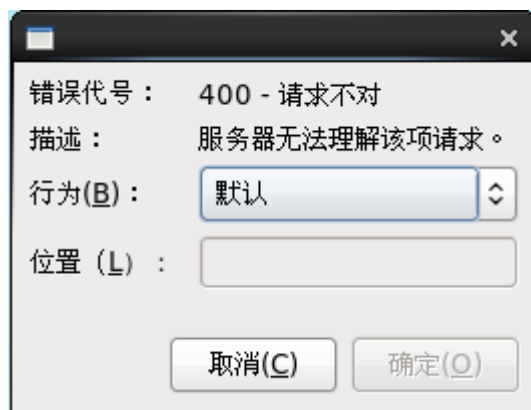


图 1-4 编辑错误信息

在图 1-4 的窗口中，【**行为**】一栏选择【**默认**】将显示缺省的简短错误信息；选择 URL 并在【**位置**】栏中输入一个包括 `http://` 的完整的 URL，将重定向用户到另一远程连接；选择【**文件**】并输入一个在网络服务器 `DocumentRoot` 下的文件名，将重定向用户到另一本地 URL。该文件名必须以一个斜划线 (/) 开始，默认对应于 `DocumentRoot` 目录下。

例如，若要将一个【**404 没有找到**】错误编码重定向到一个名为 `404.html` 的文件中，先将 `404.html` 粘贴到 `Documentroot/errors/404.html`。在这里，`DocumentRoot` 是已定义的 `DocumentRoot` 目录（缺省是 `/var/www/html`）。然后，对【**404 没有找到**】错误编码选择【**文件**】作为行为，并在【**位置**】处输入

/errors/404.html。

回到图 1-3，在【默认错误页页脚】菜单下，有如下几种选择：

【显示页脚和电子邮件地址】在所有错误页面的底部，除显示缺省的 Apache 脚注外，同时显示在 ServerAdmin 指令中指明的网络维护人员的 email 地址。

【显示页脚】在错误页面的底部，仅显示缺省的 Apache 脚注。

【无页脚】不显示任何脚注。

1.1.2.2 记录日志

在缺省状态下，Apache 将传输日志写入文件/var/log/httpd/access_log，而将错误日志写入文件/var/log/httpd/error_log。



图 1-5 日志

【**传输日志**】包含了一个记录所有连接服务器的尝试的列表。该列表记录了用户的 IP 地址、尝试连接的时间及日期，以及尝试连接的文件名。输入该文件的路径及文件名。如果该路径及文件名不是以斜划线 (/) 开始，则该路线是相对于服务器根目录的，该选项对应于 TransferLog 指令的内容。

在【**使用定制记录设施**】上打勾，并在【**定制日志字串**】栏中输入一个自定义记录字符串，则可配置一个自定义记录格式。该项对应于 LogFormat 指令的内容。如需要关于 LogFormat 指令的格式的详细信息，可参考 http://httpd.apache.org/docs/mod/mod_log_config.html#formats。

【**错误日志**】包含了一个记录所有服务器错误的列表。输入该文件的路径及文件名。如果该路径及文件名不是以斜划线 (/) 开始，则该路线是相对于服务器根目录的。该选项对应于 ErrorLog 指令的内容。

使用【**日志级别**】选择框可以设定错误信息的详细程度。可设定的等级从简到详分为：紧急情况、警告、严重、错误、提醒、注意、信息和调试 (emerg, alert, crit, error, warn, notice, info or debug)。该选项对应于 LogLevel 指令的内容。

选择【**逆向 DNS 查寻**】菜单将定义 HostnameLookups 指令的内容。选择【**无逆向查寻**】将设置此选项为关闭。选择【**逆向查寻**】将设置此选项为打开。选择【**双重逆向查寻**】将设置此选项为双向。

如果选择【**逆向查寻**】，服务器将自动解析所有连接用户所使用 IP 的对应主机名。也就是说，服务器将对 DNS 建立一个或多个连接，以找到该 IP 所对应的主机名。

如果选择【**双重逆向查寻**】，服务器将进行一个双向查找。也就是说，在服务器进行了如上所述的一次逆向查找以后，它将对查找的结果进行一次正向查找。在正向查找的 IP 地址中，至少应该有一个等于原先进行的逆向查找的结果。

一般情况下，该选项应该被设置成【**无逆向查寻**】，否则 DNS 要求将使服务器的负载增大，而使得服务器速度下降。当服务器较忙时，以上影响将尤其明显。

不光是对于服务器，逆向寻找和双向寻找对于整个网络来说也有相似的问题。因此，无论是为了您的服务器，还是为了整个网络，请将此选项选为【**无逆向查寻**】。

1.1.2.3 环境

Apache 可以使用 mod_env 模块来对环境变量进行配置。环境变量被用于向 CGI（Common Gateway Interface，通用网关接口）脚本和 SSI 页面传递数据。可以使用【环境】页面来配置该 Apache 模块的指示。



图 1-6 环境变量

使用【为 CGI 脚本设置】栏以设置将向 CGI 脚本和 SSI 页面传输数据的环境变量。例如：若要将环境变量 MAXNUM 设置成 50，可点击位于【环境变量】栏中的【添加】按钮，并在弹出的菜单窗口【环境变量】栏中输入 MAXNUM，在【要设立的值】栏中输入 50，然后点击【确认】。【环境变量】栏对应于 SetEnv

指令的内容。

使用**【传递给 CGI 脚本】**栏以在 Apache 首次启动时将环境变量的数值传送给 CGI 脚本。在 shell 提示中输入指令 `env` 可以看见此环境变量。点击位于**【传递给 CGI 脚本】**栏中的**【添加】**按钮并输入环境变量的名称，然后点击**【确认】**。

【传递给 CGI 脚本】栏对应于 `PassEnv` 指令的内容。

使用**【为 CGI 脚本取消设置】**栏以删除一个环境变量。点击位于**【为 CGI 脚本取消设置】**栏中的**【添加】**按钮，并输入环境变量的名字将重置该变量。该栏对应于 `UnsetEnv` 指令的内容。

1.1.2.4 性能

使用**【性能】**页面以配置文件夹的选项。该页面对应于 `<Directory>` 指令中的内容。

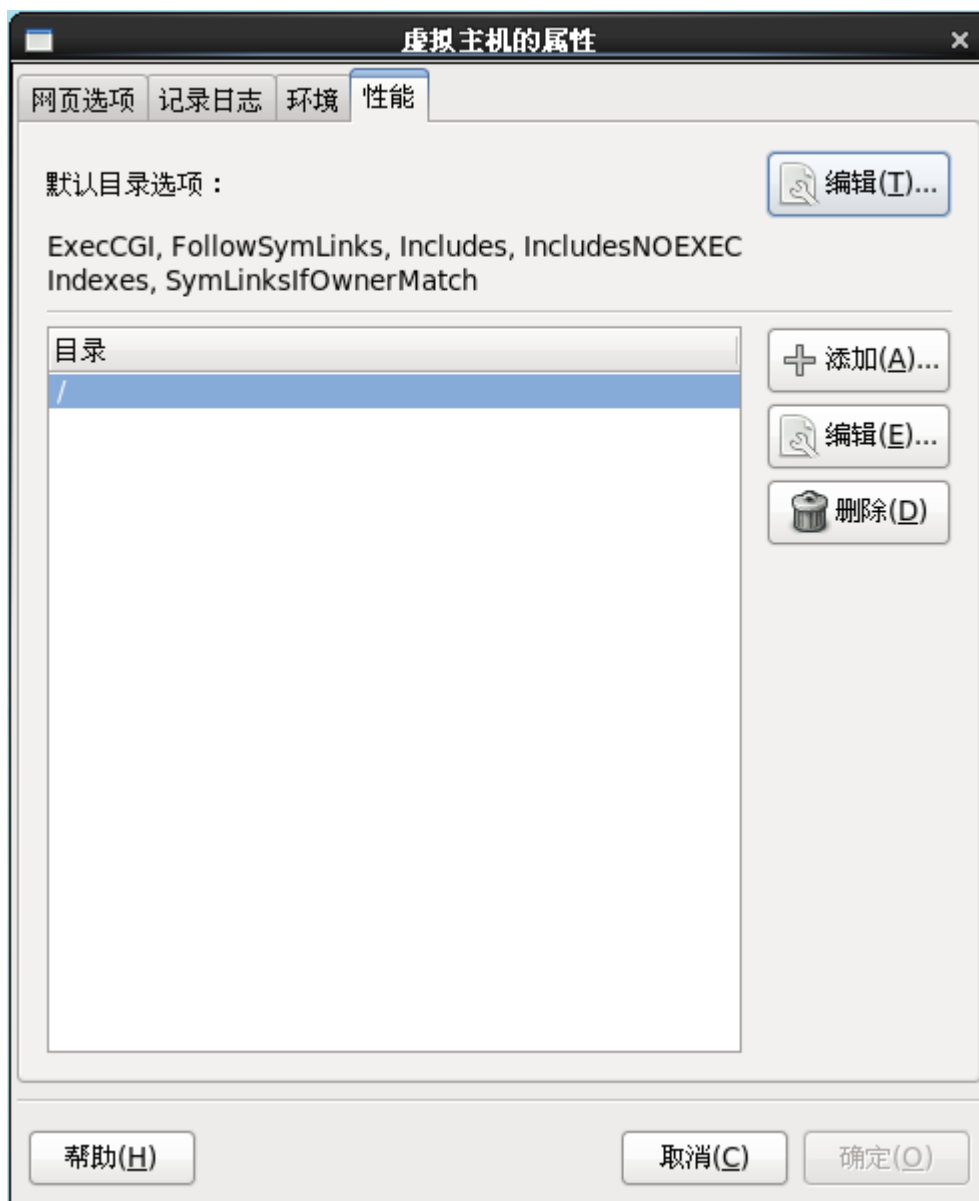


图 1-7 性能

点击位于右上角的【编辑】按钮以对【默认目录选项】进行配置。该配置对所有未被【目录】列表指定的文件夹起作用。可选择的选项在<Directory>指令中的 Options 指令已被列出。可配置的选项包括：

【ExecCGI】允许 CGI 脚本运行。除非该选项被选中，CGI 脚本将不会运行。

【FollowSymLinks】允许跟随符号连接。

【Includes】允许服务器包含。

【IncludesNOEXEC】允许服务器包含，但关闭 CGI 脚本中的#exec 和

#include 指令。

【Indexes】在被要求的文件夹中不存在 DirectoryIndex(如 index.html)的情况下，显示一个包含文件夹中所有内容的列表。

【Multiview】支持协商内容 multiviews。该选项的缺省选择是关。

【SymLinksIfOwnerMatch】仅在目标文件或文件夹与连接所有人相同的情况下，跟随符号连接。

如要指定某文件夹的选项，点击位于**【目录】**列表旁的**【添加】**按钮。一个类似图 1-8 的窗口将会弹出。在位于底部的**【目录】**栏中输入文件夹名称，在右边选择需要设置的选项，然后在左边对**【顺序】**指令进行配置。该次序指令控制着判断允许或者拒绝命令的判断顺序。在**【允许列表】**的**【允许主机来自】**和**【拒绝列表】**的**【拒绝主机来自】**栏中，可以指定以下选择：

【允许所有主机】输入“全部”以允许访问所有主机。

【部分域名】允许访问所有主机名符合或以指定字符串结尾的主机。

【完全 IP 地址】允许访问到指定的 IP 地址。

【子网】如 192.168.1.0/255.255.255.0。

【一个网络 CIDR 说明】如 10.3.0.0/16。

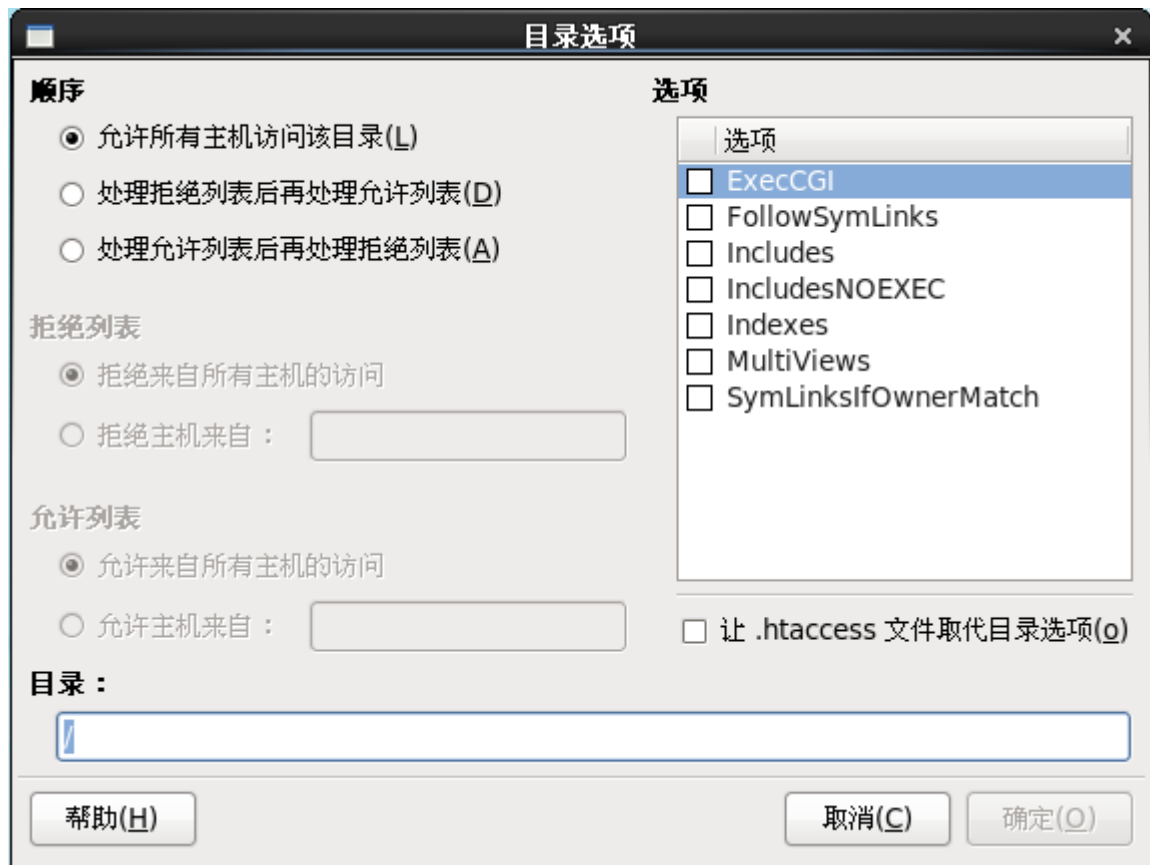


图 1-8 目录选项

如果在【让.htaccess 文件取代】目录【选项】上打勾，则 htaccess 文件夹中的配置指令将拥有优先权。

1.1.3 虚拟主机设置

您可以使用【HTTP 配置工具】来配置虚拟主机。虚拟主机使您可以在同一台机器上运行数个拥有不同 IP 地址、不同主机名、或不同端口的服务器。例如，您可以使用虚拟主机在同一个 Apache 服务器上同时支持 `http://www.your_domain.com` 和 `http://www.your_second_domain.com` 两个网站。对于缺省虚拟主机或者基于 IP 虚拟主机来说，该选项对应于<VirtualHost>指令中的内容。对于基于名字的虚拟主机来说，该选项对应于<NameVirtualHost>指令中的内容。

Apache 为虚拟主机设置的指令只会对该虚拟主机有效。如果一个指令是 siver-wide 并且是通过使用【编辑默认设置】按钮设置的，当该指令不存在于虚拟主机的设置中时，该指令将会使用它的缺省设置。例如，在【主要设置】标签中您可以定义一个【网主电子邮件地址】，但并不指定每个虚拟主机相对应的单个 email 地址。【HTTP 配置工具】包括了一个如图 1-9 的默认虚拟主机。

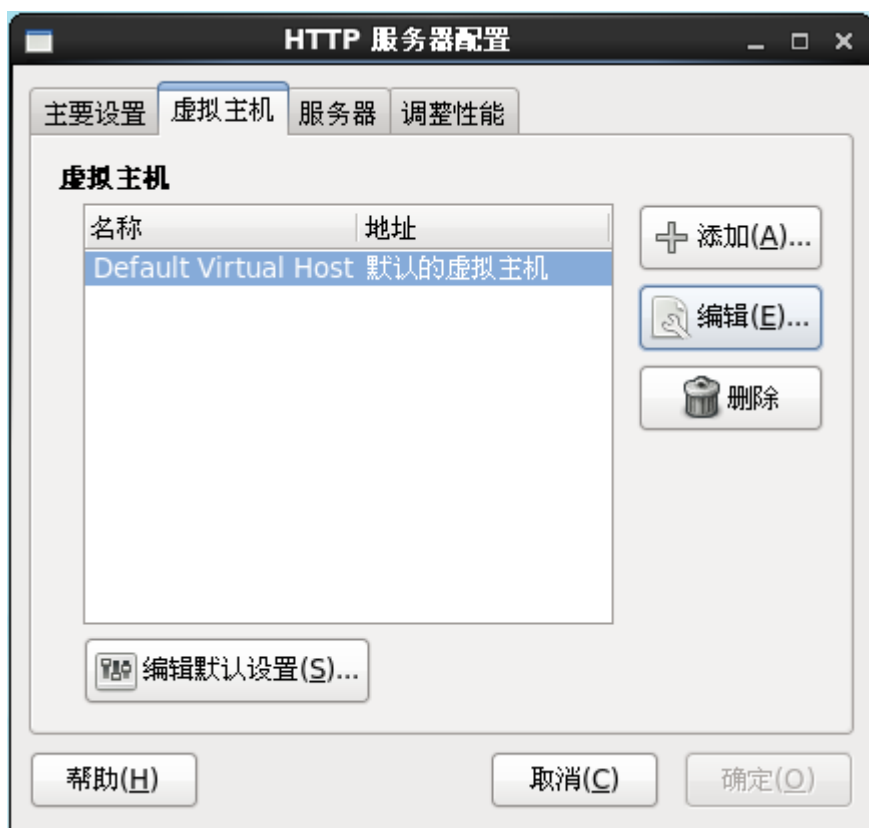


图 1-9 虚拟主机设置

在您机器上的 Apache 说明文件或者在 <http://www.apache.org/docs/vhosts/> 都提供了更多有关虚拟主机的资料。

1.1.3.1 添加或编辑一个虚拟主机

点击【**虚拟主机**】标签然后点击【**添加**】按钮以添加一个虚拟主机，一个如图 1-10 的窗口将弹出。您也可以从列表选择一个主机，然后点击【**编辑**】按钮来编辑它。



图 1-10 添加一个虚拟主机

1.1.3.2 常规选项

【常规选项】的设置仅对正在被配置的虚拟主机有效。

在**【虚拟主机名】**栏里对虚拟主机名进行设置。Apache 配置工具将使用该名称来区别不同的虚拟主机。将虚拟主机的**【文档根目录】**设置为拥有根文件（如 index.html）的目录。该选项对应于 VirtualHost 指令中的 DocumentRoot 指令的内容。**【网主电子邮件地址】**对应于 VirtualHost 指令中的 ServerAdmin 指令内容。如果您选择在错误页面显示脚注及 email 地址，该 email 将在错误页面的脚注中显示出来。

在【主机信息】栏内，选择【默认虚拟主机】、【基于 IP 的虚拟主机】或【基于名称的虚拟主机】。

1) 默认虚拟主机

如果您选择【默认虚拟主机】，如图 1-11 将出现，您仅需要配置一个缺省虚拟主机。当被要求的 IP 地址并没有在另外的虚拟主机中被明确列出的时候，缺省虚拟主机的设置将会被用到。如果缺省虚拟主机未被定义，则主服务器设置将会被用到。



图 1-11 使用默认虚拟主机

2) 基于 IP 虚拟主机

如果您选择**【基于 IP 虚拟主机】**，如图 1-12 将出现。根据服务器的 IP 地址，在该窗口可以配置<VirtualHost>指令中的内容。在**【IP 地址】**栏中指定此 IP 地址。如要指定若干个 IP 地址，各个地址之间可用空格来分隔。如需要指定端口，可使用“IP 地址：端口”的结构。使用“：*”以配置该 IP 地址的所有端口。在**【服务器主机名称】**栏中指定虚拟主机的主机名。



图 1-12 基于 IP 的虚拟主机设置

3) 基于名称的虚拟主机

如果您选择**【基于名称的虚拟主机】**，如图 1-13 将会出现。根据服务器的主机名，在该窗口可以配置 NameVirtualHost 指令中的内容。在**【IP 地址】**栏中

指定此 IP 地址。如要指定若干个 IP 地址，各个地址之间可用空格来分隔。如需要指定端口，可使用“IP 地址：端口”的结构。使用“：*”以配置该 IP 地址的所有端口。在【主机名】栏中指定虚拟主机的主机名。在【别名】栏中，点击【添加】按钮可以增加一个主机名别名。该别名对应于 NameVirtualHost 指令中的 ServerAlias 指令的内容。



图 1-13 基于名称的虚拟主机设置

1.1.3.3 SSL

 注意：在使用 SSL 的同时不能使用基于名称的虚拟主机。这是因为在 HTTP 名称虚拟(用以识别适当的基于名称的虚拟主机)执行之前，SSL 同步(当浏览器接收安全网络服务器

的证书的时候)已经执行了。如果您希望使用基于名称的虚拟主机, 则它将只能在非安全网络服务器下正常工作。

如果一个 Apache 服务器没有配置支持 SSL, 则该服务器和用户之间的信息将是未经加密的。对于不包含有个人或机密资料的网站来说, 这是合适的设置。例如, 一个用于分发开放源代码软件及说明文件的开放源代码网站就没有将信息加密的必要。但是, 一个需要信用卡帐号的电子商业网站就应该使用 Apache 的 SSL 支持方式以对它的信息进行加密。激活 Apache 的 SSL 支持方式将激活 mod_ssl 安全模块。如要将其激活, 首先需要使用 **【HTTP 配置工具】** 在 **【主要设置】** 标签下的 **【可用地址】** 中, 设置允许使用端口 443 的权限。接下来, 在 **【虚拟主机】** 标签中, 选择虚拟主机名, 然后点击 **【编辑】** 按钮。从上方的菜单中选择 **【SSL】** 然后在 **【启用 SSL 支持】** 选项上打勾, 如图 1-14。在 **【SSL 配置】** 部分已经使用虚拟数字证书进行了事先配置。数字证书被用来向您的安全网络服务器提供证明, 并帮助客户端的网络浏览器对其进行识别。您必须购买自己的数字证书, 不要使用虚拟数字证书。

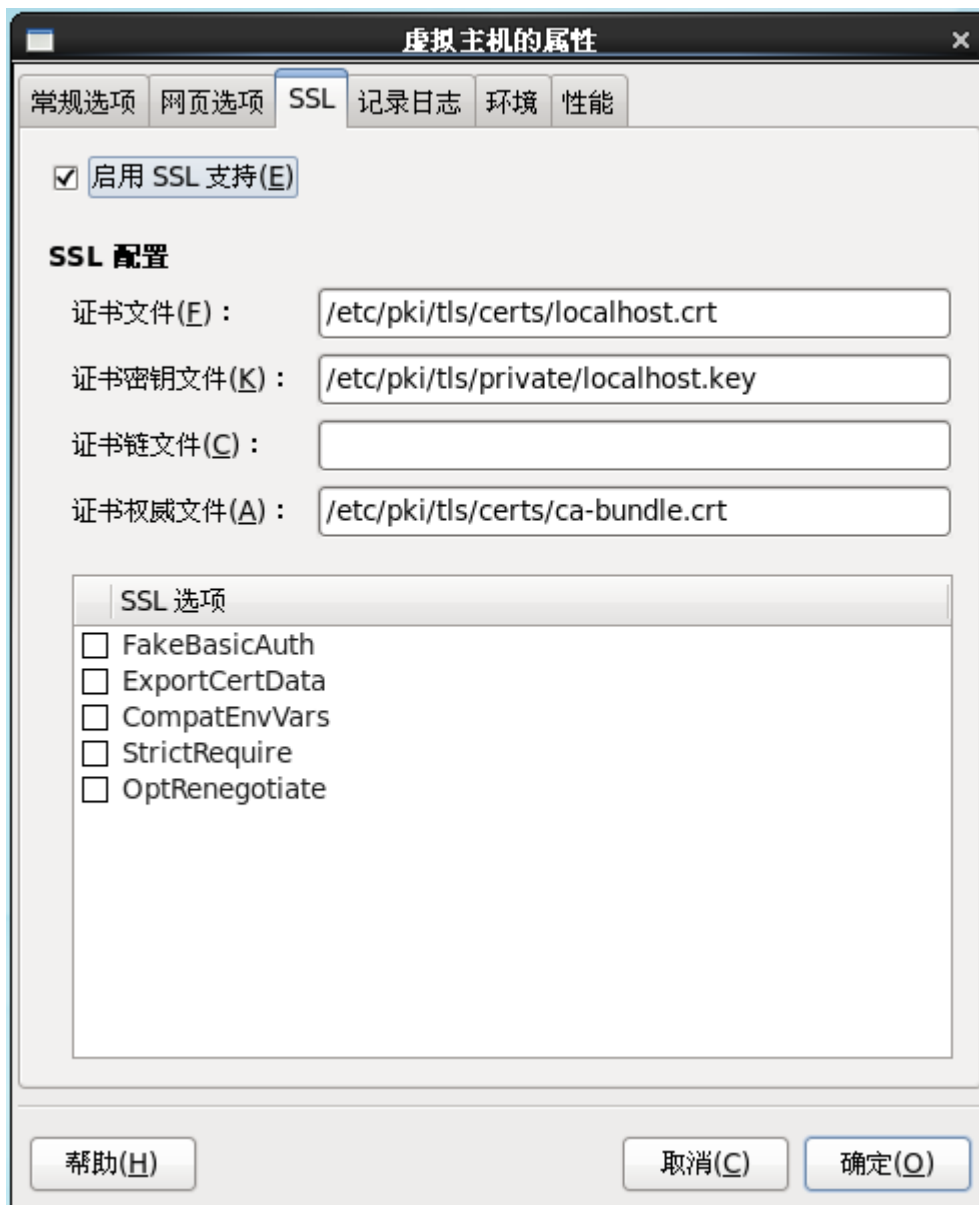


图 1-14 SSL 支持

1.1.3.4 其它虚拟主机

虚拟主机的【网页选项】、【记录日志】、【环境】和【性能】标签与点击【编辑默认设置】按钮时出现的选项相同。唯一的区别是这些选项现在只对正在进行配置的单个虚拟主机有效。关于更详细的资料，请参考虚拟主机的缺省设置部分。

1.1.4 服务器设置

在【服务器】标签下可以对基本的服务器设置进行配置。在绝大多数情况下，设置可以使用他们的缺省值。



图 1-15 服务器设置

【**锁文件**】的值对应于 LockFile 指令中的内容。该指令将编译 Apache 时使用的 lockfile 的路径设为 USE_FCNTL_SERIALIZED_ACCEPT 或 USE_FLOCK_SERIALIZED_ACCEPT，它必须被保存在本地硬盘上。除非日志目录被放在一个 NFS 共享上，否则它的各缺省值不应被修改。而当日志目录被放在一个 NFS 共享上时，缺省值应该被更改为本地硬盘上的某个位置，且只有 root 用户可见可读。

【**PID 文件**】的值对应于 PidFile 指令中的内容。该指令规定了用于存储服务器过程 ID (pid) 的文件。该文件应仅能被 root 用户读取。在绝大多数情况下，该值的缺省值不应被改动。

【**核心转储目录**】的值对应于 CoreDumpDirectory 指令中的内容。Apache 在转储核心之前将尝试切换至该文件夹。该数值的缺省值是 ServerRoot。但是如果服务器上的用户无法写入该文件夹的话，核心转储将无法被写入。如果您希望能写入核心转储以进行调试，需要该值更改为一个用户可写入的目录。

【**用户**】的值对应于 User 指令中的内容。该值设置了服务器用来回复请求

的用户名。该用户的设置决定了服务器的访问。网站的访问者将无法访问到任何该用户无法访问到的文件。该值的缺省值是 `apache`。

该用户所拥有的特权应仅被限于所有外界可见的文件。该用户同时也将是服务器产生的所有 CGI 程序的属主。该用户没有权力执行任何不属于 HTTP 请求的应答的代码。

 注意：除非有必要，否则不要将该用户设置为 `root` 用户。给予 `root` 用户该用户的权力将给网络服务器造成很大的安全漏洞。

在通常情况下，父 `httpd` 进程以 `root` 用户来运行，但将会马上传给 `apache` 用户。因为服务器需要捆绑到一个 1024 以下的端口，服务器必须以 `root` 用户启动。小于 1024 的端口是为系统运行保留的，所以除 `root` 用户以外，其他用户无法使用。但是当服务器已经将自己捆绑到一个端口以后，服务器会在接收任何连接请求之前把程序传给 `apache` 用户。

【组群】值对应于 `Group` 指令中的内容。组指令与用户指令很相似。它规定了服务器用于应答请求的组名。该值的缺省值也是 `apache`。

1.1.5 调整性能

点击**【调整性能】**标签以配置子服务进程的最高限，以及配置客户连接 `apache` 选项。在绝大多数情况下，缺省设置的值不需要改动。改变这些设置将有可能影响到网络服务器的性能。



图 1-16 调整性能设置

将【**最多连接数量**】设置服务器可应答客户并发请求的最大数量。每一个连接将会创建一个子 httpd 进程。在进程数达到该值以后，其他用户将无法连接到服务器，直到一个子服务进程重新释放为止。在不重新编译 Apache 的情况下，该值最大不能超过 256。该选项对应于 MaxClients 指令中的内容。

【**连接超时**】定义了（以秒为单位）浏览器在接收和发送信息的过程中所应等候的时间。特别的，连接超时定义了服务器在接收到 GET 请求时所应等待的时间，在接到 POST 或 PUT 请求并等候接收 TCP 信息包时所应等待的时间，以及在等待 ACKs 对 TCP 信息包作出应答时所应等待的时间。该值的缺省值是 300 秒。在绝大多数情况下，该值是适中的。该选项对应于 Timeout 指令中的内容。

将【**每次连接最多请求数量**】设置为一个持续连接所允许的最大请求数。该值的缺省值为 100。在绝大多数情况下，该值是适中的。该选项对应于 MaxRequestsPerChild 指令中的内容。

当您选择了选项【**允许每次连接可有无限请求**】，并把 MaxKeepAliveRequests 指令设置为 0，则一个持续连接所允许的最大要求数不被限制。

如果没有选择【允许持久性的连接】选项，则 KeepAlive 指令的值被设置成假。如果选择了该选项，则 KeepAlive 指令被设置成真，而 KeepAliveTimeout 指令的值被设置成【下次连接的超时时间】的值。该指令设置了以秒为单位，服务器将等待下一个请求的时间。一旦请求已经被接收了，则将使用【连接超时】的值。

将【允许持久性的连接】的值设置的高，在有较多用户尝试连接服务器的情况下，将有可能使服务器速度减慢。该值越高，则最后一个客户所需要等待下一个连接的服务进程等待时间也越长。

1.1.6 保存设置

如果您不希望保存您的 Apache 配置设置，点击位于【HTTP 配置工具】右下的【取消】按钮。系统提示您确认这个决定。如果您点击【是】，则您的设置将不会被保存。

如果您希望保存您的配置设置，点击位于右下方的【确定】按钮。如图 1-17 的对话框将会弹出。如果选择【是】，则您的设置将会被存入/etc/httpd/conf/httpd.conf。注意您原来的配置文件将会被覆盖。

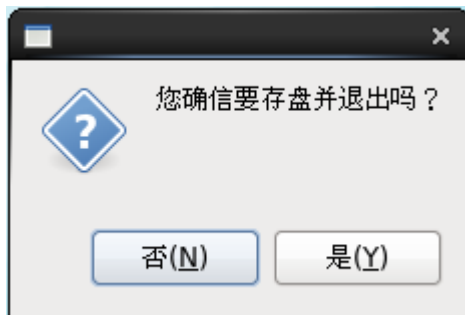


图 1-17 保存并退出



注意：在保存了您的设置以后，必须使用指令 `service httpd restart` 来重启 Apache。您必须用 root 用户登录执行此指令。

1.1.7 其它资源

关于 Apache 的更多资料，请参考以下资源。

Apache 文档—如果您已安装 apache-manual 包并正在运行 Apache 网络服务器 daemon(httpd)，您可以看到 Apache 文档。打开一个浏览器，并在正运行 Apache 的服务器上输入 URL：<http://usr/share/doc/httpd-2.0.46>。

<http://www.apache.org>—The Apache Software Foundation。

<http://httpd.apache.org/docs/>—Apache HTTP Server Version 1.3 User's Guide。

Apache:The Definitive Guide by Ben Laurie and Peter Laurie; O'Reilly & Associates, Inc。

1.2 Apache 命令和模块

对于大多数用户来说，Apache 的缺省配置能够保证正常工作。您可能不需要更改任何 Apache 的配置选项。但如果您的确需要更改某个缺省配置选项，您将需要了解部分选项的作用以及它们所在的位置。本节将包含对您有用的配置选项。

 注意：如果您准备使用中标麒麟服务器操作系统软件提供的 GUI 工具 Apache 配置工具，请不要手工编辑 ApacheWeb 服务器的 httpd.conf 配置文件。反之，如果您希望手工编辑 httpd.conf，则不要使用 Apache 图形配置工具。关于 Apache 配置工具的更多资料，请参考 Official 中标麒麟服务器操作系统自定义指南。

在您安装了 apache 包以后，ApacheWeb 服务器的说明文件将被存于 http://your_domain/manual/ 或者您也可以在 <http://httpd.apache.org/docs/> 找到该说明文件。ApacheWeb 服务器的说明文件包括了一个对所有 Apache 配置选项完整描述的完全列表。为了您的方便起见，本节对中标麒麟服务器操作系统提供的 Apache 版本中用到的配置指令进行简短描述。

在阅读 Web 服务器的配置文件时，请注意它同时包括了安全和非安全 Web 服务器。安全 Web 服务器在一个虚拟主机上运行，它的配置被存在 httpd.conf 文件中。

1.2.1 启动和终止 httpd

httpd RPM 会安装/etc/init.d/httpd 程序，该程序可通过/sbin/service 命令调用。

输入以下指令以启动服务器：

```
/sbin/service httpd start
```

如果您把 Apache 作为一个安全服务器运行，则您会被询问您的密码。在输入密码后，服务器将被启动。

输入以下指令以终止服务器：

```
/sbin/service httpd stop
```

restart 指令是一个终止然后启动服务器的简便方法。该 **restart** 指令明确的终止，然后启动您的服务器。所以如果您使用的是安全服务器的话，您将被询问密码。**restart** 指令的格式如下：

```
/sbin/service httpd restart
```

如果您刚刚完成了对 **httpd.conf** 文件的编辑，您不需要终止并启动服务器。您可以使用 **reload** 指令。当您使用 **reload** 时，您将不需要输入密码。您的密码将会在 **reload** 过程中被记住。但当您使用终止或启动时，则密码不会被记忆。**reload** 指令的格式如下：

```
/sbin/service httpd reload
```

设定机器开机时缺省启动 **httpd** 服务，**httpd** 过程将自动启动。如果您在使用安全服务器，您将在开机后被询问密码，除非您为您的安全服务器生成了一个没有密码保护的密匙。

1.2.2 httpd.conf 中的配置命令

Apache Web 服务器的配置文件是 **/etc/httpd/conf/httpd.conf**。该 **httpd.conf** 文件已被加上了足够多的注释而显得较通俗易懂。对于大多数人来说，它的缺省设置将能够正常工作，因此您可能不需要更改 **httpd.conf** 中的选项。但您可能会希望熟悉其中最重要的一些配置选项。

如果您需要配置 Apache，您只需编辑 **httpd.conf** 文件，然后 **reload**，或重新启动 **httpd** 进程。

在开始编辑 **httpd.conf** 文件以前，请先将原来的文件存为 **httpd.confold**（或其他任何您希望的名字）。这样如果您在编辑过程中出现了错误，您仍有一个原配置文件的备份。

如果您在编辑中出错使您的 Web 服务器不能正常工作，首先检查在 **httpd.conf** 文件中您刚刚修改过的内容。保证您没有打错任何东西。下一部要检查的应该是 Web 服务器的错误日志(**/var/log/httpd/error_log**)。依您的经验多少的

不同, 该错误日志不一定会很容易看懂。但如果您刚刚发现了一个问题, 错误日志中的最后一个条目应该会为出错的原因提供一些资料。

下面将按存在于文件中的次序向您提供各选项的简短介绍。这些介绍将不是完整的。如果您需要更多的资料, 请参考 <http://httpd.apache.org/docs-2.0/> 的 Apache 组说明文件。

关于 `mod_ssl` 指令的更多资料, 请参考 http://httpd.apache.org/docs-2.0/mod/mod_ssl.html 的 `mod_ssl` 用户手册。

1.2.2.1 ServerRoot

`ServerRoot` 是用来保存服务器文件的最高级的目录。无论是安全还是非安全服务器该值都被设置为 `/etc/httpd`。

1.2.2.2 PidFile

`PidFile` 存储了服务器用于记录它的进程 ID(pid) 的文件的文件名。Web 服务器的设置是 `/var/run/httpd.pid`。

1.2.2.3 Timeout

`Timeout` 定义了(以秒为单位的)您的服务器在通讯时等待接收和传送的时间。更准确地说, `Timeout` 定义了服务器接收 GET 请求的等待时间, 接收 POST 或 PUT 请求的 TCP 包的等待时间, 以及 ACK 响应 TCP 包的等待时间。`Timeout` 的缺省值是 60 秒。在大多数情况下, 该设置将是合适的。

1.2.2.4 KeepAlive

`KeepAlive` 的设置规定了服务器是否允许持续连接(或者说, 是否允许每个连接作出超过一个请求)。`KeepAlive` 可以被用来防止任何用户过多的使用服务器的资源。缺省设置中, `KeepAlive` 的设置是 Off, 也就是说不允许持续连接。关于相关的限制每个连接的请求数量的办法, 请参考章节 1.2.2.5 `MaxKeepAliveRequests`。

1.2.2.5 MaxKeepAliveRequests

该指令设置了一个持续连接所能做出的最大请求数。Apache 项目组建议使用一个高的设置, 以提供服务器的性能。该设置的缺省设置是 100。在大多数情况下应该是适当的。

1.2.2.6 KeepAliveTimeout

KeepAliveTimeout 设置了服务器在回复了一个要求后用于等待下一个要求的时间。如果等待时间超过该时间而未收到新的要求，该连接将被关闭。当收到了新的要求后，**Timeout** 指令将开始计时。

1.2.2.7 IfModule

`<IfModule>`和`</IfModule>`标签之间包含条件指令。如果标签`<IfModule>`中的模块被编译入了 Apache 服务器，则这些指令将被处理。或者，如果一个`!`被包含在模块名前，则只有当模块没有被编译入 Apache 服务器时，这些指令将被处理。

`mod_mime_magic.c` 文件被包括在这些 `IfModule` 标签中。`mod_mime_magic` 模块与 UNIX 文件指令相似。它检查文件内容的开头部分并使用"魔术数字"以及其他提示以决定文件的 **MIME** 类型。

如果 `mod_mime_magic` 模块被编译入了 Apache，这些 `IfModule` 标签将告诉 `mod_mime_magic` 模块用于包含提示定义的文件路径。在这个例子中，该文件为 `/usr/share/magic`。

`mod_mime_magic` 模块在缺省下是不能被编译的。如果您想使用它，请参考章节 1.2.4 添加模块到服务器。

1.2.2.8 Mem-Sptcific Sever-Pool 命令

在 Apache HTTP Server 2.0 中，管理 `server-pool` 的任务由一组称为 **MPM** 的模块完成。`server-pool` 由于所基于的 **MPM** 的不同而具有不同的特性。因此，需要一个 `IfModule` 容器来定义某个 `server-pool` 所使用的 **MPM**。

在缺省情况下，Apache HTTP Server 2.0 用 `prefork` 和 `worker` **MPM** 定义 `server-pool`。以下是 **MPM-specific server-pool containers** 中的一些指令：

1) StartSevers

StartServers 设置了在开始的时候，生成服务器进程的数量。因为网络服务器将根据通信负载动态地杀死或生成服务器进程，该设置无需改变。该数值的缺省值在 `prefork` **MPM** 中是 8，在 `worker` **MPM** 中是 4。

2) MaxRequestPerChilds

MaxRequestsPerChilds 设置了在子服务器死亡前该服务器子进程可以服务的

请求的最大数值。该设置的主要功能是防止生存时间过长的进程造成内存溢出。该数值的缺省值在 `prefork MPM` 中是 4000，在 `worker MPM` 中是 0。

3) MaxClients

`MaxClients` 设置了在同一时间可以运行的服务器进程总数的极限(或者说，同时连接的客户端的极限)。该设置的主要功能是防止一个失控的 `Apache HTTP` 服务器拖垮您的操作系统。对于一个比较繁忙的服务器，该值需要被设置的高一些。无论对于哪个 `MPM`，缺省值都是 300。但是，在使用 `prefork MPM` 时，该值不要超过 256。

4) MinSpareServers 和 MaxSpareServers

这两个值仅应用于 `prefork MPM`。`ApacheWeb` 服务器通过通信量来保持一定数量的备用服务器进程，以动态地适应检测到的负载。服务器将检查正等待请求的服务数量。如果该数量超过 `MaxSpareServers`，则终止一部分服务；如果少于 `MinSpareServers`，则生成一部分服务。

服务器的缺省 `MinSpareServers` 是 5；缺省 `MaxSpareServers` 是 20。在几乎所有的情况下，这两个缺省数值都应该是合适的。不要将 `MinSpareServers` 设置得太大，那样即使在通讯量很少的情况下，服务器的进程负荷也会很大。

5) ThreadsPerChild

该值仅用于 `worker MPM`，它设置每一子程序的线程数，缺省值为 25。

1.2.2.9 Listen

`Listen` 指令用于指定 `Web` 服务器用来接收请求的端口。`Web` 服务器的设置是：端口 80 用于非安全 `Web` 通讯而(在定义安全服务器的虚拟主机标签中)端口 443 用于安全 `Web` 通讯。

如果您将 `Apache` 用于接收的端口设置为 1024 以下，您必须是 `root` 用户才能启动它。

对于 1024 或以上的端口，普通用户可以正常启动 `httpd`。

`Listen` 也可以用来指定服务器将接受连接的特定 `IP` 地址。

1.2.2.10 Include

`Include` 允许其它的配置文件在运行状态下被包含进来，这些配置文件的路径可以是对于 `ServerRoot` 的相对路径，也可以是绝对路径。

当服务器使用如 `mod_ssl`, `mod_perl`, 和 `php` 等一些独立模块的时候, 在全局环境的配置文件 `httpd.conf` 中务必包含下面的命令:

```
Include conf.d/*.conf
```

1.2.2.11 LoadModule

`LoadModule` 用来装载动态共享对象(DSO)模块。关于 Apache 的 DSO 支持的更多资料, 包括怎样使用 `LoadModule` 指令, 请参考 1.2.4 添加模块到服务器一节。注意, 在 Apache HTTP Server 2.0 中, 模块的先后次序已经不再重要。

1.2.2.12 ExtendedStatus

`ExtendedStatus` 指令控制着当调用 `server-status` 管理器时, Apache 是生成基本服务器状态信息(当该值为关时)还是生成详细服务器状态信息(当该值为开时)。使用 `Location` 标签调用 `server-status`。关于调用 `server-status` 的更多资料, 请参照 1.2.2.58 `Location` 一节。

1.2.2.13 IfDefine

`<IfDefine>`和`</IfDefine>`标签包含了当在`<IfDefine>`标签中测试为正确的情况下将会被应用的配置指令。如果测试为错误, 这些指令将不起作用。

在`<IfDefine>`标签中的测试将是一个一个参量(例如, `HAVE_PERL`)。如果该参量已被定义, 或者说它已被作为一个参数提供给了服务器的启动指令, 则该测试为正确。在此情况下, 当网络服务器启动时, 测试为正确, 因此在 `IfDefine` 标签中的指令将被运用。

根据缺省设置, `<IfDefine HAVE_SSL>`标签将包含用于安全服务器的虚拟主机标签。`<IfDefine HAVE_SSL>`标签同样也包含了用于 `ssl_module` 的 `LoadModule` 和 `AddModule` 指令。

1.2.2.14 User

`User` 指令将设置服务器用于响应请求所用的用户名。`User` 的设置决定了服务器的访问权限。任何该用户无法访问的文件同样将无法被您的网站的访问者看到。该值的缺省设置是 `apache`。

该用户应该仅拥有足够的权限以访问所有允许被外界访问者看见的文件。该用户同时也是服务器产生的所有 `CGI` 进程的所有者。该用户不应该被允许执行

任何不应响应给 HTTP 请求的代码。

除非有必要，否则不要将该用户设置为 root 用户。这样做将使 Web 服务器出现很大的安全漏洞。

在正常运转下，父 httpd 过程将以 root 用户来运行，但将会马上传给 apache 用户。因为服务器需要 bind 到一个 1024 以下的端口(安全网络通讯的缺省端口是 443; 非安全网络通讯的端口是 80)，服务器必须作为 root 用户启动。小于 1024 的端口是仅为系统运行保留的，所以除 root 用户以外，其他用户无法使用。但是当服务器已经将自己附加到一个端口以后，服务器会在接收任何连接请求之前把进程传给 apache 用户。

1.2.2.15 Group

Group 指令和 User 指令类似。Group 规定了服务器用于响应请求的组名。该值的缺省值也是 apache。

1.2.2.16 ServerAdmin

ServerAdmin 应该设置为 Web 服务器管理人员的 email 地址。该 email 地址将会在服务器生成的错误信息 Web 页中显示出来，以便用户通过发邮件向管理员报告问题。ServerAdmin 的缺省被设为 root@localhost。

一个设置 ServerAdmin 的好方法是将其设为 webmaster@your_domain.com。然后在/etc/aliases 内将 webmaster 别名设置为 Web 服务器的负责人。最后，运行 /usr/bin/newaliases 以添加该 alias。

1.2.2.17 ServerName

您可以使用 ServerName 以设置（不同于主机真实名称的）服务器主机名。例如，当您的服务器真实名称是 foo.your_domain.com 时，您可以将主机名设置为 www.your_domain.com。注意 ServerName 必须是一个有效、您有权使用的的域名服务（DNS）名（不要随便编造一个名字）。

下面是一个设置 ServerName 的指令：

```
ServerName www.your_domain.com:80
```

如果您指定一个 ServerName，需要确认已经您将它的一对 IP 地址和服务器名包括在了/etc/hosts 文件中。

1.2.2.18 UseCanonicalName

UseCanonicalName 的缺省设置是 off, 当服务器接到需要指向自己的请求时, 它将使用来自客户请求中所包含的值来指示自己。当它被设置为 On 时, UseCanonicalName 允许服务器使用 ServerName 和 Port 建造一个指向自己的 URL。当服务器在回复客户请求指向自己时, 它将使用该 URL。

1.2.2.19 DocumentRoot

DocumentRoot 是用来包括绝大多数用于回复请求的 HTML 文件的文件夹。该值的缺省值(无论对于安全或非安全网络服务器来说)是/var/www/html。例如, 服务器可能会收到如下的要求:

```
http://your_domain/foo.html
```

服务器将在缺省文件夹中找寻如下文件:

```
/var/www/html/foo.html
```

如果您希望更改 DocumentRoot 以便使安全和非安全 Web 服务器不共享同一文件夹, 请参见“使用虚拟主机”一节。

1.2.2.20 Directory

<Directory/path/to/directory>和</Directory>标签之间包含仅对此文件夹及其所有子文件夹中起作用的配置指令。任何可使用于一个文件夹的指令都可以用在<Directory>标签中。<File>标签可以用同样的方法被使用以应用于单个的文件上。

根据缺省值, 非常有限的参数可以作用于根目录之上, 只使用了 Option (见“Options”一节) 和 AllowOverride (见“AllowOverride”一节) 指令。在该配置之下, 任何在系统中需要更多权限的文件夹必须在各自的设置段中给出明确的设定。

通过使用 Directory 标签, DocumentRoot 可以拥有更宽松的参数, 以提供 HTTP 请求的回复。

cgi-bin 文件夹通过 ExecCGI 选项, 可以设置为允许运行 CGI 脚本。如果您需要在其他的文件夹中运行 CGI 脚本, 您需要为相应的目录进行 ExecCGI 设置。

例如，如果您的 `cgi-bin` 是 `/var/www/cgi-bin`，但您需要在 `/home/my_cgi_directory` 文件夹中运行 CGI 脚本，在 `httpd.conf` 文件的 `Directory` 指令中添加一个 `ExecCGI` 指令如下：

```
<Directory /home/my_cgi_directory>
Options+ExecCGI
</Directory>
```

要允许 CGI 脚本在 `/home/my_cgi_directory` 中运行，除了设置 `ExecCGI` 以外，您还需要其他几个步骤。您需要将 `AddHandler` 指令非注释化以便将拥有 `.cgi` 扩展名的文件鉴别为 CGI 脚本。关于设置 `AddHandler` 的方法，请参见“`AddHandler`”一节。CGI 脚本以及脚本的完整路径的权限，必须被设为 `0755`。最后，脚本的拥有者和文件夹的拥有者必须是同一个用户。

1.2.2.21 Options

`Options` 指令控制着对每一个特定文件夹有效的服务器功能。例如，在为根目录设置的限制性参量下，`Options` 被仅仅设置为 `FollowSymLinks`。除了允许服务器在根目录中跟从符号链接以外，没有任何功能被激活。

缺省设置，在 `DocumentRoot` 文件夹中，`Options` 被设置为包括 `Indexes`，`Includes` 以及 `FollowSymLinks`。`Indexes` 允许服务器在没有指定 `DirectoryIndex`（如 `index.html`）的情况下生成一个文件夹列表。`Includes` 的意思是允许 `server-side-includes`。`FollowSymLinks` 允许服务器跟从该文件夹中的符号链接。

如果您希望虚拟主机能够识别这些 `Options` 的话，您也将需要在虚拟主机指令中包含选项声明。

例如，由于在 `<Directory"/var/www/html">` 指令栏中的 `Options Includes` 行的关系，`server-side-includes` 在 `/var/www/html` 文件夹中已被激活。但是，如果您希望一个虚拟主机承认在 `/var/www/html` 中 `server-side-includes` 是被允许的，在虚拟主机标签中需要包含以下一段：

```
<Directory /var/www/html>
Options Includes
</Directory>
```

1.2.2.22 AllowOverride

`AllowOverride` 指令将设置是否任何 `Options` 可以被一个 `.htaccess` 文件的声明

所覆盖。按照缺省，根目录和 DocumentRoot 文件夹都是不允许覆盖的。

1.2.2.23 Order

Order 指令控制着 allow 和 deny 指令执行评估的顺序。缺省配置是在 DocumentRoot 文件夹中先执行 Allow 指令再执行 Deny 指令。

1.2.2.24 Allow

Allow 指定了哪些请求者有权访问某一特定文件夹。该请求者可以是 all，一个域名，一个 IP 地址，一个部分 IP 地址，一个 network/netmask 对等等。您的 DocumentRoot 文件夹的缺省配置是 Allow from all。

1.2.2.25 Deny

Deny 类似于 Allow，不同的只是它是指定哪些请求者无权访问某一特定文件夹。DocumentRoot 文件夹的缺省配置是不拒绝任何人的请求。

1.2.2.26 UserDir

UserDir 是在每个用户的主目录下用于存放可被 Web 服务器提供的个人 HTML 文件的子目录名。该值的缺省是 public_html。例如，服务器有可能收到如下要求：

```
http://your_domain/~username/foo.html
```

服务器将寻找以下文件：

```
/home/username/public_html/foo.html
```

在以上例子中，/home/username 是用户的主目录(注意用户主目录的缺省路径在您的系统中有可能会不同)。

确认用户主目录的权限被正确设置。用户的主目录的权限必须被设置为 0755。读位(r)和执行位(x)必须在用户的 public_html 文件夹上被设置(可以使用 0755)。在用户 public_html 文件夹中会被使用的文件必须至少设置为 0644。

1.2.2.27 DirectoryIndex

DirectoryIndex 是当用户通过在文件夹名后加入斜划线(/)以要求该文件夹索引时服务器提供的缺省页面。

当用户要求 http://your_domain/this_directory/页面时，如果 DirectoryIndex 页

面存在，他们将得到该页面。如果该页面不存在，他们将得到一个服务器生成的文件夹列表。DirectoryIndex 的缺省是 index.html、index.html.var。服务器将尝试寻找以上任何文件，并打开第一个找到的文件。如果服务器未能找到任何以上文件而 Options Indexes 在该文件夹已被设置，则服务器将生成并回复一个包含该文件夹下所有子文件夹及文件的 HTML 格式的列表。

1.2.2.28 AccessFileName

AccessFileName 包含了在各文件夹中服务器用来管理访问权限资料的文件名。该值缺省设置是.htaccess。如果该文件存在，它将管理所在文件夹的访问权限资料。

在 AccessFileName 指令之后，由一组 Files 标签负责管理任何以.ht 开始的文件的访问权限。由于安全原因，这些指令将拒绝对于.htaccess 文件(和其他以.ht 开始的文件)的任何 Web 访问要求。

1.2.2.29 CacheNegotiatedDocs

按照缺省，服务器将要求代理服务器不要缓存任何在基本内容之上通过交互所得的文件(或者说，可能因为输入不同而导致反馈不同的文件)。如果您未注掉 CacheNegotiatedDocs，则该选项将被禁用，而代理服务器将被允许缓存此类文件。

1.2.2.30 TypesConfig

TypesConfig 指定了 MIME 类型对照(文件扩展名对应于内容类型)的缺省列表的文件名。该缺省文件是/etc/mime.types。当添加 MIME 类型对照时，建议的做法是使用 AddType 指令，而不是编辑/etc/mime.types 文件。

1.2.2.31 DefaultType

DefaultType 设置了当 MIME 类型不明的时候，网络服务器使用的缺省内容类型。缺省值假设为纯文本内容类型。

1.2.2.32 HostnameLookups

HostnameLookups 可以被设为 on, off, 或 double。如果您通过设置其为开而允许 HostnameLookups，服务器将自动解析所有连线用户所使用的 IP 对应的主机名。也就是说，服务器将对 DNS 建立一个或多个连接，以找到该 IP 所对应的主机名。

如果您设置该值为 double，服务器将进行一个双向查找。也就是说，在服务

器进行了如上所述的一次逆向查找以后，它将对查找的结果进行一次正向查找。在正向查找的 IP 地址中，至少应该有一个是等于之前进行的逆向查找的。

在一般情况下，该选项应该被设置成 off，否则 DNS 请求将使服务器的负载增大，而使得服务器速度下降。当服务器较忙时，以上影响将尤其明显。

不光是对于服务器，逆向查找和双向查找对于整个网络来说也有相似的问题。因此，无论是为了您的服务器，还是为了整个网络，请将该值设置为 off。

1.2.2.33 ErrorLog

ErrorLog 包含了记录服务器错误文件的文件名。正如该指令所指出的，网络服务器的错误记录文件将被存于 /var/log/httpd/error_log。

如果网络服务器生成了任何错误而您不清楚错误出自何处，错误日志文件将是一个很好的检查错误的地方。

1.2.2.34 LogLevel

LogLevel 设置了错误日志中错误信息的详尽程度。从最简单到最详尽，该值包括了紧急情况、警告、严重、错误、提醒、注意、信息和调试 (emerg, alert, crit, error, warn, notice, info or debug) 几个级别。该值的缺省设置是 warn。

1.2.2.35 LogFormat

在 httpd.conf 文件中的 LogFormat 指令设置了访问日志中的消息格式。希望该格式能使您的访问日志更易读。

LogFormat 命令配置各种 web 服务器日志文件的格式，它的使用同时依赖 CustomLog 命令中的设置。

当 CustomLog 设置为 combined 时，有如下的格式选项：

1) %h(远程主机的 IP 地址或主机名)

列出正在发出请求的远程主机的 IP 地址；若 HostnameLookups 的设置为 on，则记录这一客户的主机名，除非在 DNS 中查找不到。

2) %l(rfc931)

无作用。在记录文件中，您将看到一个“-”以代替它。

3) %u(authenticated user)

如果需要验证，该值将被用户用以验证自己的用户名。一般情况下，该值将不会被用到，而您将看到一个“-”以代替它。

4) %t(date)

请求的日期和时间。

5) %r(request string)

浏览器或客户端送来的请求字符串。

6) %s(status)

回复客户端的 HTTP 状态编码。

7) %b(bytes)

文档大小。

8) %\%"%{Referer}i\%(referrer)

列出将客户主机作为 Web 服务器的网页 URL 地址。

9) %\%"%{User-Agent}i\%(user-agent)

列出发出请求的 web 浏览器的类型。

1.2.2.36 CustomLog

CustomLog 指定访问日志文件及其格式。在 Web 服务器的缺省配置中，CustomLog 定义了访问日志文件为：/var/log/httpd/access_log。如果您需要生成一个基于访问的服务器性能统计，您将需要知道此文件的位置。

CustomLog 的缺省格式为 combined，例：

```
remotehost rfc931 user date "request" status bytes referrer user-agent
```

1.2.2.37 ServerSignature

ServerSignature 指令会在任何服务器生成的文件中(例如，送回客户的错误信息)添加一行包含有 Apache 服务器版本及服务器名的信息。ServerSignature 的缺省被设置为 On。您可以将其设置为 off，以取消该签名行。您也可以将其设置为 EMail。EMail 将在签名行中添加一个 mailto:ServerAdmin 的 HTML 标签。

1.2.2.38 Alias

Alias 设置允许将文件夹放在 DocumentRoot 目录以外而仍能被 Web 服务器访问。任何以别名结尾的 URL 都将自动解析到别名的真实路径。根据缺省，一个别名已经被设置。一个 icons 目录可以被 Web 服务器访问到，但它并不在 DocumentRoot 下。该 icons 目录，是/var/www/icons/的别名，而不是/var/www/html/icons/。

1.2.2.39 ScriptAlias

ScriptAlias 设置定义了 CGI 脚本(或其他任何类型的脚本)的位置。一般来说,您不希望将 CGI 脚本放入 DocumentRoot 下。如果 CGI 脚本被放入 DocumentRoot 下,它们将有可能被作为文本文件看到。即使您并不反对其他用户看见(并使用)您的 CGI 脚本,展示它们的结构将有可能给不道德的访问者用以探索脚本文件中存在的安全漏洞的机会,并对服务器的安全产生威胁。根据缺省, cgi-bin 文件夹是/cgi-bin/的一个 ScriptAlias,并存在于/var/www/cgi-bin/。

/var/www/cgi-bin 文件夹已设置了 ExecCGI 选项。也就是说,在该文件夹下的 CGI 脚本的运行已被允许了。

1.2.2.40 Redirect

当网页移动时, Redirect 被用于重新定位到新的 URL, 格式如下:

```
Redirect /<old-path>/<file-name> http://<current-domain>/<current-path>/<file-name>
```

其中<old-path>是<file-name>的原来路径,<current-domain>和<current-path>是<file-name>的当前域和路径。

依据上面的命令,任何有关<file-name>的请求都会被重新定位到新的路径。

使用 Apache HTTP 服务器包含的 mod_rewrite 模块,可以获得更高级的重定位技术。更多关于配置 mod_rewrite 模块的信息,可参见 Apache Software Foundation 的在线文档 http://httpd.apache.org/docs-2.0/mod/mod_rewrite.html。

1.2.2.41 IndexOptions

IndexOptions 通过添加图标和文件描述来控制服务器生成的目录列表的外观。如果 Options Indexes 被设置(参见 1.2.2.21Options),当 Web 服务器接收到一个如下的 HTTP 请求后,网络服务器将生成一个目录列表: http://your_domain/this_directory/。首先,您的 Web 服务器在该目录中搜寻一个 DirectoryIndex 列表中指定的文件(通常是 index.html)。如果网络服务器无法找到这些文件,它将生成一个列有该目录下所有子目录和文件的 HTML 页面。使用 httpd.conf 中的一部分指令(包括 IndexOptions),您可以修改该目录列表的外观。

您的缺省配置将 FancyIndexing 设置为开。如果 FancyIndexing 为开,点击列表每列的列顶将使文件按该相应顺序排序。在同一列顶上再按一次,将使文件逆

序排序。FancyIndexing 同时也将根据文件扩展名对不同的文件显示不同的图标。如果您使用 AddDescription 指令并将 FancyIndexing 设为开，在服务器生成的文件夹列表中每个文件将包括一个简短描述。

IndexOptions 还有一些其他的可以用来控制服务器生成目录外观的参量。这些参量包括 IconHeight 和 IconWidth，以使服务器在生成的页面中包含图标的 HTML HEIGHT 和 WIDTH 标签；IconsAreLinks 可以被用来使图标和文件名一样指向一个 HTML 连接，等等。

1.2.2.42 AddIconByEncoding

该指令指定在服务器生成的目录列表中使用 MIME 编码文件所对应的图标。例如，根据缺省，在服务器生成的目录列表中，服务器在 MIME 编码的 x-compress 和 x-gzip 文件旁显示了 compressed.gif 图标。

1.2.2.43 AddIconByType

该指令指定在服务器生成的目录列表中 MIME 类型文件所对应的图标。例如，在服务器生成的目录列表中，在 mime 类型是“text”的文件旁，将显示 text.gif 图标。

1.2.2.44 AddIcon

AddIcon 告诉服务器在服务器生成的目录列表中特定文件扩展名所对应的图标。例如，服务器在以.bin 或.exe 为扩展名的文件旁，将显示 binary.gif 图标。

1.2.2.45 DefaultIcon

DefaultIcon 包含了在服务器生成的目录列表中没有对应图标的文件所应显示的图标。根据缺省，unknown.gif 图形文件将对应于这些文件。

1.2.2.46 AddDescription

在服务器生成的目录列表中，您可以使用 AddDescription 以对您指定的部分文件显示描述性的文字(同时您需要激活 IndexOptions 中的 FancyIndexing)。您可以通过指定文件名，通配符表达式，或文件扩展名来指定您希望该指令所作用的文件。例如，您可以使用如下行：

```
AddDescription "A file that ends in .ni" .ni
```

在服务器生成的文件夹列表中，所有文件扩展名为.ni 的文件将在文件名后

拥有一个“A file that ends in .ni”的描述。注意您需要将 FancyIndexing 设为开。

1.2.2.47 ReadmeName

ReadmeName 指定(当它存在于该目录时)将附于服务器生成的目录列表最后的文件的文件名。Web 服务器将首先尝试将该文件作为一个 HTML 文件附在最后, 然后尝试将该文件作为一个纯文本文件附在最后。ReadmeName 的缺省值是 README.html。

1.2.2.48 HeaderName

HeaderName 之前(当它存在于该目录时)将附于服务器生成的目录列表开始的文件的文件名。和 ReadmeName 相似, Web 服务器将首先尝试将该文件作为一个 HTML 文件附加, 然后(如果失败)尝试将该文件作为一个纯文本文件附加。

1.2.2.49 IndexIgnore

IndexIgnore 列出文件扩展名、部分文件名、通配符表达式和完整文件名。Web 服务器在 Web 服务器生成的目录列表中不会包括任何符合以上变量的文件。

1.2.2.50 AddEncoding

AddEncoding 给出了应该用于指定某些特定编码类型的文件扩展名。AddEncoding 也可以被用来要求一些浏览器(不是所有的都可以)对某些下载的文件进行解压缩。

1.2.2.51 AddLanguage

AddLanguage 将文件扩展名对应于特定的内容语言。该指令在内容协商时非常有用, 服务器根据客户的浏览器选择语言, 从数个文件中决定回复一个文件。

1.2.2.52 LanguagePriority

LanguagePriority 允许您设置提供文件时, 不同语言的优先级。该设定将会在客户未在他们的浏览器中设置语言偏好时生效。

1.2.2.53 AddType

AddType 命令用以定义或覆盖一个缺省 MIME 类型及其文件扩展名。下面的指令将让 Apache HTTP 服务器识别.tgz 文件扩展名:

```
AddType application /x-tar .tgz
```

1.2.2.54 AddHandler

AddHandler 将文件扩展名对应于特定的处理程序。例如，`cgi-script handler` 可以被用来对应于 `.cgi` 文件扩展名，以自动将所有以 `.cgi` 结尾的文件视为 CGI 脚本。即使对于 `ScriptAlias` 文件夹之外的文件，只要您跟随这里的说明，此功能也可正常工作。

在 `httpd.conf` 文件中，您有一个 CGI AddHandler 行：

```
AddHandler cgi-script .cgi
```

您需要将该行非注释化。然后 Apache 将对(即使不在 `ScriptAlias` 之内的)以 `.cgi` 结尾的文件运行 CGI 脚本。根据缺省，`ScriptAlias` 将把 `/cgi-bin/` 目录定位于 `/var/www/cgi-bin/`。

您同时也需要将任何包含有 CGI 脚本的文件夹的 `Options` 中添加 `ExecCGI` 设置。关于为目录进行 `ExecCGI` 设置，请参考 1.2.2.20 `Directory` 一节。另外，您需要确定 CGI 脚本以及包含 CGI 脚本的目录的权限均被正确设置。CGI 脚本以及脚本的目录全路径必须被设为 `0755`。最后，目录和脚本文件的所有人必须相同。

如果您使用虚拟主机并希望他们能够识别 `ScriptAlias` 之外的 CGI 脚本，您需要添加同一个 `AddHandler` 行至您的虚拟主机设置中。

除了 CGI 脚本以外，您的 Web 服务器也可以使用 `AddHandler` 以处理服务器解析 HTML 和 `imagemap` 文件。

1.2.2.55 Action

`Action` 允许您指定一个 MIME 内容类型和 CGI 脚本对照，以使每当该种媒体类型的文件被请求时，该指定 CGI 脚本被执行。

1.2.2.56 ErrorDocument

按照缺省，在出现错误或问题时，Web 服务器将输出一个简单的(通常含义模糊)错误信息给客户端。如果不希望使用缺省，您可以使用 `ErrorDocument` 来配置 Web 服务器以使其输出一个自定义的信息或重定向客户端至一个本地和外地的 URL。

`ErrorDocument` 指令只是简单的将一个 HTTP 回复编码对应于一个将送回客

户的信息或 URL。

需要输出的信息必须用一对双引号（“ ”）括起来。

1.2.2.57 BrowserMatch

BrowserMatch 指令允许服务器来定义环境参量，或根据标识客户浏览器的用户代理 HTTP header 字段，作出适当的处理。按照缺省，Web 服务器使用 **BrowserMatch** 来拒绝任何有已知问题的浏览器的连接并对已知的不能使用 keepalives 和 HTTP header 刷新的浏览器禁用这些功能。

1.2.2.58 Location

<Location> 和 **</Location>** 标签用于生成一个可通过 URL 来控制访问的 container。

例如，如需允许从内部服务器域登录的用户看到状态报告，可使用下面的命令：

```
<Location /server-status>
SetHandler server-status
Order deny,allow
Deny from all
Allow from <.example.com>
</Location>
```

其中<.example.com>是 Web 服务器的二级域名。

如想向来自域内部的请求提供服务器的配置报告(包括已安装的模块和其配置命令)，可使用下面的命令：

```
<Location /server-info>
SetHandler server-info
Order deny,allow
Deny from all
Allow from <.example.com>
</Location>
```

1.2.2.59 ProxyRequests

将 Apache HTTP 服务器同时配置成为一个代理服务器，您可以采用如下操作：去掉<IfModule mod_proxy.c>行、ProxyRequests、<Proxy>节的每一行前面的井号(#)；设置 ProxyRequests 为 On；并在<Proxy>节中使用 Allow from 命令设置

允许访问的域。

1.2.2.60 ProxyVia

ProxyVia 指令决定了一个 HTTP 一个 Via:header 行是否同通过 Apache 代理服务器的请求或是回复一同发送。如果 ProxyVia 被设置为 On, Via:header 行将显示主机名; 设置为 Full 将显示主机名和 Apache 版本; 设置为 Off, 所有的 Via 将在传送过程中不能被更改; 设置为 Block, Via:行会被删除。

1.2.2.61 Cache 指令

Apache HTTP 服务器的缺省配置文件中有很多添加了注释的 cache 命令, 以下介绍其它一些比较重要的关于 cache 配置的命令。

1) CacheEnable

指定 cache 的类型 (磁盘、内存, 或是文件描述的 cache); 缺省情况下, CacheEnable 将磁盘的根目录或下层目录作为 URL 的 cache。

2) CacheRoot

指定包含有 cache 文件的目录名, 缺省值为/var/httpd/proxy/。

1.2.2.62 NameVirtualHost

对于任何您设置的基于名称虚拟主机的 IP 地址和端口号, 您需要使用 NameVirtualHost 指令。当您希望为不同的域设置不同的虚拟主机, 但是您没有足够的(或不愿使用)IP 地址来对应于每一个由您的 Web 服务器提供服务的域名时, 您需要使用基于名称虚拟主机配置。

基于名称的虚拟主机不能作为安全服务器运行, 如需使用虚拟主机作为安全服务器, 请采用基于 IP 地址的虚拟主机。

为启动基于名称的虚拟主机, 请使用 NameVirtualHost 配置命令, 并加入正确的 IP 地址, 然后为每一个虚拟主机添加 VirtualHost containers。

1.2.2.63 VirtualHost

<VirtualHost>和</VirtualHost>标签包含任何针对于虚拟主机的配置指令。绝大多数配置指令可在虚拟主机标签内使用, 并仅对该虚拟主机有效。

httpd.conf 中提供被注释掉的 VirtualHost container, 里面包括一些最基础的设置虚拟主机所需的命令。

缺省的 SSL 虚拟主机 container 被移入了/etc/httpd/conf.d/ssl.conf 文件。

1.2.2.64 SSL 设置指令

/etc/httpd/conf.d/ssl.conf 文件中的命令用于激活使用 SSL 和 TLS 进行安全 Web 通信。

SetEnvIf 根据进入连接的标题头设置环境变量。它不是一个单独的 SSL 指令，而是用在/etc/httpd/conf.d/ssl.conf 文件中。它在上下文中的含义是：停止 HTTP 的 keepalive 设置，而允许 SSL 去关闭那些未从客户端浏览器发来关闭通知的连接。这对于一些没有以安全可靠的方式关闭 SSL 连接的浏览器客户端是非常必要的。

关于 SSL 配置文件中的更多命令，请参考：

http://localhost/manual/mod/mod_ssl.html

http://httpd.apache.org/docs-2.0/mod/mod_ssl.html

对于大多数情况，SSL 命令在安装时已经被恰当地配置，请在改变 Apache HTTP 安全服务器的配置时格外注意，因为任何错误的配置都将可能使得服务器容易遭受攻击。

1.2.3 缺省模块

Apache HTTP 已经发布了许多模块，缺省情况下安装并启用了如下一些 httpd 包：

```
mod_access.so
mod_auth.so
mod_auth_anon.so
mod_auth_dbm.so
mod_auth_digest.so
mod_include.so
mod_log_config.so
mod_env.so
mod_mime_magic.so
mod_cern_meta.so
mod_expires.so
mod_headers.so
mod_usertrack.so
mod_unique_id.so
mod_setenvif.so
```

```
mod_mime.so
mod_dav.so
mod_status.so
mod_autoindex.so
mod_asis.so
mod_info.so
mod_dav_fs.so
mod_vhost_alias.so
mod_negotiation.so
mod_dir.so
mod_imap.so
mod_actions.so
mod_speling.so
mod_userdir.so
mod_alias.so
mod_rewrite.so
mod_proxy.so
mod_proxy_ftp.so
mod_proxy_http.so
mod_proxy_connect.so
mod_cache.so
mod_disk_cache.so
mod_file_cache.so
mod_mem_cache.so
mod_cgi.so
```

另外，通过安装附加的包，还可以获取以下几个模块：

```
mod_auth_mysql
mod_auth_pgsq
mod_perl
mod_python
mod_ssl
php
```

1.2.4 添加模块到服务器

因为 Apache 支持 DSO，您可以很容易的在 Web 服务器中装载 Apache 模块或编译自己的模块。支持 DSO 的意义在于各模块可以在运行时被装载。由于这

些模块将仅在需要时被装载，在被装载前它们并不会使用任何内存，因此总的来说所需内存将降低。

Apache 公司在 <http://httpd.apache.org/docs-2.0/dso.html> 提供完整的 DSO 文档。或者，如果您安装了 http-manual 包，也可以在这里找到相关文档：
<http://localhost/manual/mod/>。

只有在/etc/httpd/conf/httpd.conf 中使用了 LoadModule 指令，Apache HTTP 服务器才能使用 DSO；如果模块是单独包的形式，还必须在/etc/httpd/conf.d/目录下的模块配置文件中加入该行命令。

如果在 http.conf 中添加或删除了模块，必须重新装载或启动 Apache HTTP 服务器。

如果新建了一个模块，请首先安装 httpd-devel 包，因为里面有在编译 DSO 时所需的包含文件、头文件，以及 Apache 扩展应用程序(/usr/sbin/apxs)。

写好一个模块后，可以使用/usr/sbin/apxs 在 Apache 资源目录外编译。关于使用/usr/sbin/apxs 的更多信息可参考 <http://httpd.apache.org/docs-2.0/dso.html> 和 apxs 手册。

编译后把模块放在/usr/lib/httpd/modules/目录下，然后在 httpd.conf 中加入：

```
LoadModule <module-name>
<path/to/module.so>
```

其中<module-name>是模块的名字，<path/to/module.so>是 DSO 的路径。

1.2.5 使用虚拟主机

如果您希望使用中标麒麟服务器操作系统提供的 GUI 实用程序 HTTP 配置工具，您不可以在 ApacheWeb 服务器上编译您自己的模块或编辑 Apache Web 服务器的 httpd.conf 配置文件。反之，如果您需要添加任何模块到 Apache 或手工编辑 httpd.conf 文件，不要使用 HTTP 配置工具。

可以使用 Apache 的虚拟主机功能以在同一机器上运行拥有不同 IP 地址、不同主机名或不同端口的服务器。如果您对使用虚拟主机感兴趣，在您机器的 Apache 说明文档或在 <http://httpd.apache.org/docs-2.0/vhosts/>上都有完整的资料。



注意：在使用 SSL 的同时不能使用基于名字的虚拟主机。这是因为在 HTTP 请求(用以识别适当的基于名字的虚拟主机)执行之前，SSL 同步(当浏览器接收安全 Web 服务器的证

书的时候)已经执行了。如果您希望使用基于名字的虚拟主机,则它将只能在非安全 Web 服务器下正常工作。

1.2.5.1 虚拟主机的建立

最好使用 httpd.conf 中的虚拟主机 container 来建立一个基于名字的虚拟主机,如下:

```
#NameVirtualHost *  
#<VirtualHost *>  
# ServerAdmin webmaster@dummy-host.example.com  
# DocumentRoot /www/docs/dummy-host.example.com  
# ServerName dummy-host.example.com  
# ErrorLog logs/dummy-host.example.com-error_log  
# CustomLog logs/dummy-host.example.com-access_log common  
#</VirtualHost>
```

为激活它,请去掉 NameVirtualHost 前的井号(#),用机器的 IP 地址替换星号(*)。然后去掉<VirtualHost>container 的注释并定制其中的命令。

去掉<VirtualHost>行前的#,用服务器的 IP 地址替换*;设置 ServerName 为机器的有效 DNS 名;配置其它一些必要的命令。

<VirtualHost>container 具有很强的自定义特性,能够使用几乎所有的主服务器设置的命令。

 提示: 如果想把一个虚拟主机配置成用于监听非缺省的端口,那么该端口必须在 /etc/httpd/conf/httpd.conf 全局配置文件中用 Listen 命令添加进去。

为启用新建的虚拟主机,必须重新装载或启动 Apache HTTP 服务器。

更多关于新建和配置基于名称虚拟主机和基于 IP 地址虚拟主机的信息请参考 <http://httpd.apache.org/docs-2.0/vhosts/>。

1.2.5.2 安全 Web 服务器虚拟主机

Web 服务器的缺省配置是运行一个非安全 Web 服务器以及一个安全 Web 服务器。这两个服务器使用相同的 IP 地址和主机名,但分别使用 80 和 443 端口,这样它们可以同时进行连接。SSL 之所以能够加强 HTTP 的传输功能,有一方面是因为它比标准的 HTTP 协议更严密,所以安全服务器每秒不能处理太多的页。因此,限制由安全服务器获取信息的大小是一个很好的办法,尤其是对于比较繁

忙的站点。

不要使用基于名称虚拟主机作为安全服务器，因为在 HTTP 尚未识别出该虚拟主机之前，SSL 的握手信号就已经发出了。基于名字虚拟主机仅适用于非安全服务器。

安全服务器的配置指令包含在/etc/httpd/conf.d/ssl.conf 文件的虚拟主机标签中。

缺省时，安全和非安全服务器使用同一个 DocumentRoot，推荐为安全 web 服务器使用不同的 DocumentRoot。

如需中止非安全服务器接收连接，请在 httpd.conf 中将 Listen 80 行注释掉，如下：

```
#Listen 80
```

有关更高级的配置技巧，您可查阅 Apache Software Foundation 的在线文档：

<http://httpd.apache.org/docs-2.0/ssl/>

<http://httpd.apache.org/docs-2.0/vhosts/>

1.2.6 附加资源

从以下资源，您可以获取更多关于 Apache HTTP 服务的信息。

1.2.6.1 有用站点

<http://httpd.apache.org/> — Apache HTTP 服务的官方网站，有包含所有命令和模块的文档。

<http://www.modssl.org/> — mod_ssl 的官方网站。

<http://www.apacheweek.com/> — 综合性的关于 Apache 的每周通讯。

1.2.6.2 相关书籍

Apache Desktop Reference（作者 Ralf S. Engelschall；Addison Wesley 出版） — Ralf S. Engelschall 是 ASF 成员，mod_ssl 的作者，该书简要但全面地介绍了 Apache HTTP 服务器在编辑、配置、运行时的方法；可从 <http://www.apacheref.com/> 获得。

Professional Apache（作者 Peter Wainwright；Wrox Press Ltd 出版） — 是 Wrox Press Ltd 出版公司 "Programmer to Programmer" 系列丛书中的一本，对于

Web 服务器管理员的初学者和高级者都可适用。

Administering Apache（作者 Mark Allan Arnold；Osborne Media Group 出版）

— 对于需要设置好的安全服务器的工作者来讲，本书是一个很好的选择。

Apache Server Unleashed（作者 Richard Bowen，et al；SAMS BOOKS 出版）

— 一本关于 Apache HTTP Server 的百科全书。

Apache Pocket Reference（作者 Andrew Ford，Gigi Estabrook；O'Reilly 出版）

— 是 O'Reilly 口袋参考书系列之一。

2 DNS 配置和使用

2.1 管理 DNS 设置

【网络配置】工具中的 DNS 标签允许您配置系统的主机名、域、名称服务器和搜索域。名称服务器用来查寻网络上的其它主机。

如果 DNS 服务器的名称要从 DHCP 或 PPPoE 中检索到（或从 ISP 中检索），则不要添加主要、次要或第三 DNS 服务器。如果主机名被动态地从 DHCP 或 PPPoE 中检索（或从 ISP 中检索），则不要改变这个主机名。

通过菜单在桌面上，点击【启动】→【系统】→【首选项】→【网络连接】，选中【System eth0】，单击【编辑】按钮，图 2-1 显示的是 DNS 标签：



正在编辑 System eth0

连接名称(I)

☐ 自动连接(A)

有线 802.1x 安全性 IPv4 设置 IPv6 设置

方法(M):

地址

地址	子网掩码	网关
10.1.83.26	255.255.252.0	10.1.80.254

DNS 服务器:

搜索域(S):

DHCP 客户端 ID:

☒ 需要 IPv4 地址完成这个连接

☒ 对所有用户可用

图 2-1 配置 DNS

如图 2-1 所示，只支持主 DNS 配置，点击【应用】使配置生效。

也可以在命令行执行命令 `system-config-network` 弹出图 2-2 所示【网络配置】界面。

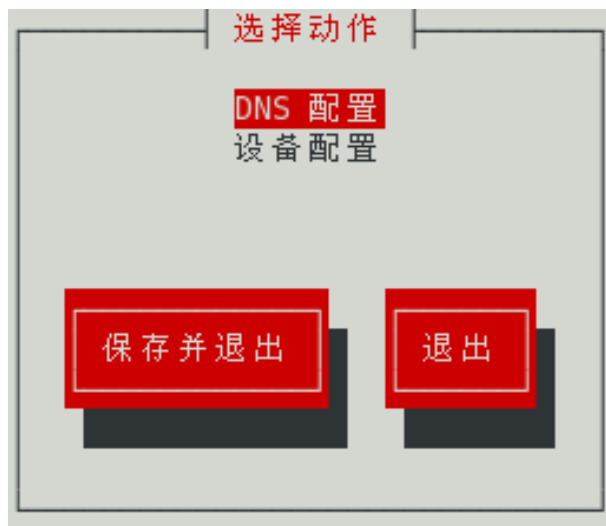


图 2-2 网络配置

如图 2-2，通过键盘的上下键选中 DNS 配置，回车，弹出图 2-3 所示 DNS 配置界面。

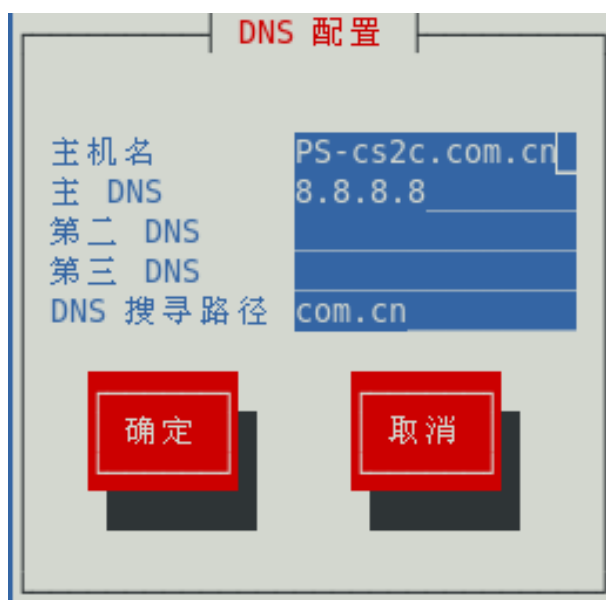


图 2-3 DNS 配置界面


注意：域名服务器部署的目的不是将系统配置成域名服务器，而是用来配置系统解析 IP 地址和主机名所用的域名服务器。

2.2 BIND 配置

本节假设您对 BIND 和 DNS 已经有最基本的理解，因此并不解释 BIND 和 DNS 的概念，只解释如何来配置 BIND 的基本服务器域。首先创建/etc/named.conf 配置文件，并且对/var/named 目录下的域配置文件进行修改。

编写/etc/named.conf 配置文件允许您增加正向主域、逆向主域和从域。

2.3 Named.conf 配置文件

named.conf 是 bind 的配置文件，当您安装了中标麒麟高级服务器操作系统后，会自动安装 bind，在/var/named/chroot/etc/目录下可以找到 named.conf 文件（须运行 system-config-bind 工具后才会生成配置文件）。它的原始内容如下所示，您可以参考相关资料对其进行修改，以建立自己所需的 DNS 服务。

```
options{
directory "/var/named";
dump-file "/var/named/data/cache_dump.db";
statistics-file "/var/named/data/named_stats.txt";
/*
* If there is a firewall between you and nameservers you want
* to talk to, you might need to uncomment the query-source
* directive below. Previous versions of BIND always asked
* questions using port 53, but BIND 8.1 uses an unprivileged
* port by default.
*/
// query-source address * port 53;
};
//
// a caching only nameserver config
//
controls {
inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
zone "." IN {
type hint;
file "named.ca";
};
```


文件第二部分 zone "."定义了根域信息，也就是当域名服务器收到域名查询时，发现客户希望查询的域名不是本地域名时，它就会查询/etc/named.ca 中定义的根域名服务器，直到最终得到希望查询的域名。

zone "localdomain"指定服务器作为 localdomain 域的主域名服务器（此处的 localdomain 是您的域名，需要根据您的实际信息修改）。file "localdomain.zone"是指定 localdomain.zone（localdomain.zone 同样是需要实际数据替换的）文件中包含所有*.localdomain 形式的域名转换数据。

文件的第三部分 zone "localhost"和 zone "0.0.127.in-addr-arpa"定义了本地回路的正反向解析，一般不修改。

文件的第四部分是关键，这里设置您需要的信息。

2.4 配置示例

下面是一个简要的示例，仅供您参考。

2.4.1 主配置文件

主配置文件，即/var/named/chroot/etc/named.conf，下面逐步分析一个比较基础的配置文件。

1) log option

```
/*
 * log option
 */
logging {
    channel default_syslog { syslog local2; severity error; };
    channel audit_log { file "/var/log/named.log"; severity error; print-time yes; };
    category default { default_syslog; };
    category general { default_syslog; };
    category security { audit_log; default_syslog; };
    category config { default_syslog; };
    category resolver { audit_log; };
    category xfer-in { audit_log; };
    category xfer-out { audit_log; };
    category notify { audit_log; };
    category client { audit_log; };
    category network { audit_log; };
}
```

```
category update { audit_log; };  
category queries { audit_log; };  
category lame-servers { audit_log; };  
};
```

这一部分是日志的设置，其中最主要的是 `file"/var/log/named.log"` 这一句指定了日志文件的位置，要正常启动 `named`，必须要保证这一文件是存在的，并且 `named` 进程对它有读写权限。

2) options

```
options {  
    directory "/etc/namedb"; //指定域名解析等文件的存放目录（须手动建立）;  
    listen-on-v6 { any; };    //支持 ipv6 的请求;  
    // If you've got a DNS server around at your upstream provider, enter  
    // its IP address here, and enable the line below. This will make you  
    // benefit from its cache, thus reduce overall DNS traffic in the Internet.  
    forwarders {  
        your.upper.DNS.address;  
    }; //指定前向 DNS，当本机无法解析的域名，就会被转发至前向 DNS 进行解析;  
    /*  
    * If there is a firewall between you and nameservers you want  
    * to talk to, you might need to uncomment the query-source  
    * directive below. Previous versions of BIND always asked  
    * questions using port 53, but BIND 8.1 uses an unprivileged  
    * port by default.  
    */  
    // query-source address * port 53;  
    /*  
    * If running in a sandbox, you may have to specify a different  
    * location for the dumpfile.  
    */  
    dump-file "/etc/named_dump.db";  
}; //指定 named_dump.db 文件的位置;
```

3) 线索域和回环域

```
// Setting up secondaries is way easier and the rough picture for this  
// is explained below.
```

```
//
// If you enable a local name server, don't forget to enter 127.0.0.1
// into your /etc/resolv.conf so this server will be queried first.
// Also, make sure to enable it in /etc/rc.conf.
zone "." {
type hint;
file "named.root";
};
zone "0.0.127.IN-ADDR.ARPA" {
type master;
file "localhost.rev";
};
```

指定线索域和本地回环域，这一部分使用一些标准的例子就可以，一般不需要修改。File "named.root"，指定该域的解析文件，其目录为 options 中 directory "/etc/namedb"，指定的。在本例中为/etc/namedb。

4) 自定义域

```
zone "test.com" {
type master; //指明该域主要由本机解析;
file "zone.test "; //指定其解析文件为 zong.test，目录为 options 中设定的目录，本例中
//为/etc/named。
}; //设定 test.com 域;
zone "0.168.192.in-addr.arpa" {
type master; //指明该域主要由本机解析;
file "zone. test.rev"; //指定其解析文件为 zong.test.rev，目录为 options 中设定的目录，本例
//中为/etc/named。
}; //指定 ipv4 地址逆向解析;
zone "4.0.0.f.0.5.2.0.1.0.0.2.IP6.ARPA" {
type master; //指明该域主要由本机解析;
allow-transfer { any; };
allow-query { any; };
file "ipv6.rev"; //指定其解析文件为 ipv6.rev，目录为 options 中设定的目录本例中
//为/etc/named。
}; //指定 ipv4 地址逆向解析;
zone "lowerlevelzone.test.com" {
```

```
type slave; //指明该域主要由低一级的域名服务器解析;
masters {
192.168.1.1;
}; //指定低一级的域名服务器的 ip 地址;
}; //设定 lowerlevelzone.test.com 域。
```

到此我们就初步建立了一个标准的 `named` 的主配置文件，接下来建立对应的域名解析或逆向解析文件。

2.4.2 域名解析

域名解析文件，即 `/etc/namedb/zone.test`。

```
@ IN SOA ns.test.com. root.test.com.(
2005030116; Serial
3600 ; Refresh
900 ; Retry
3600000 ; Expire
3600 ) ; Minimum
IN NS ns.test.com
;
ns IN A 192.168.0.1
www6 IN AAAA 2001:250:f004::10
www IN A 192.168.0.2
```

本文件前半部分是一些默认的参数设置，只需把域名改成对应您要设置的域就行，其余的不用过分深究，如果读者有兴趣可以查阅相关的手册文档。



注意： `IN NS ns.test.com;` 这一条必须有，用于指定本域的域名服务器；域名必须以 `."` 结尾。

本文件的第二部分（倒数三行），指定了该域上的主机：`ns IN A 192.168.0.1`。
`ns` 为主机名，`A` 代表地址类型为 `IPV4` 地址，`192.168.0.1` 是实际 `ip` 地址，这一条记录的含义是 `ns.test.com` 的 `ip` 地址为 `192.168.0.1`。

`www6 IN AAAA 2001:250:f004::10` — `www6` 为主机名，`AAAA` 代表地址类型为 `IPV6` 地址，`2001:250:f004::10` 是其 `IPV6` 地址，这条记录的含义是 `www6.test.com` 的 `ip` 地址是 `2001:250:f004::10`。

2.4.3 IP 地址逆向解析

`ipv4` 逆向解析：`/etc/namedb/zone.test.rev`。



图 2-4 图形化 bind 配置界面

选中【DNS 服务器】，点击菜单栏上的【新建】，选择【网络区域】。

弹出的【新网络区域】中，网络区域类型选择【master】，Class 选择【IN 互联网】，点击下方【确定】按钮，来源类型选择【正向】，点击下方【确定】按钮，输入 DNS 名为“testdns.”。弹出确认信息框，点击【确定】保存。



图 2-5 新网络区域

单击选中【DNS 服务器】，点击菜单栏上的【新建】，选择【网络区域】。弹出的【新网络区域】中，网络区域类型选择【master】，Class 选择【IN 互联网】，

点击【确定】，来源类型选择【IPv4 逆向】，点击【确定】，依次添加网络区域来源为“10.1.81”。弹出确认信息框，点击【确定】。



图 2-6 添加新网络区域

配置工具显示窗口中，单击选中 testdns，右键选择【添加】→【A IPv4 地址】。弹出的对话框中选择【选择 IPv4 前缀】菜单下的【10.1.81】，并在 IPv4 地址的最后一位填写“61”，选中【创建逆向映射记录】，点击【确定】。

点击菜单栏【保存】选项，提示信息，选择【是】。



图 2-7 AIPv4 地址

修改/etc/resolv.conf，注释掉原来的信息，增加“nameserver 10.1.81.63”将自己的主机做为 DNS 服务器。

```
[root@localhost 未命名文件夹]# service network restart
正在关闭接口 eth0: 设备状态: 3 (断开连接)

关闭环回接口:
弹出环回接口:
[root@localhost 未命名文件夹]# service named restart
停止 named:
启动 named:
[root@localhost 未命名文件夹]# nslookup testdns
Server:      127.0.0.1
Address:     127.0.0.1#53

Name:   testdns
Address: 10.1.81.61

[root@localhost 未命名文件夹]# nslookup 10.1.81.61
Server:      127.0.0.1
Address:     127.0.0.1#53

61.81.1.10.in-addr.arpa name = testdns.

[root@localhost 未命名文件夹]#
```

图 2-8 重启网络服务

启动或重启 named 服务:

```
service named restart
```

查询 named 服务的状态:

```
service named status
```

执行命令 nslookup testdns 进行正向解析。

执行命令 nslookup 10.1.81.61 进行反向解析。

3 SAMBA 配置和使用

Samba 使用 SMB 协议跨网络连接共享文件和打印机。支持此协议的操作系统包括 Microsoft Windows（通过其网上邻居）、OS/2 和 Linux。

如果有一个既有 Windows 又有 Linux 机器的网络，Samba 是有用的。Samba 允许网络中的所有系统共享文件和打印机。

3.1 配置 Samba 服务器

默认的配置文件（/etc/samba/smb.conf）允许用户把他们的主目录作为 Samba 共享来查看。它还把为系统配置的打印机作为 Samba 共享打印机来共享。换一句话说，您可以在您的系统上连接打印机，然后从网络上的 Windows 机器来打印。

3.1.1 图形化配置

要使用图形化界面来配置 Samba，使用 Samba 服务器配置工具。

Samba 服务器配置工具是用来管理 Samba 共享、用户、以及基本服务器设置的图形化工具。它修改/etc/samba/目录中的配置文件。

要使用该程序，您必须具备 root 用户权限。要从桌面启动 Samba 服务器配置工具，点击面板上的【启动】→【系统】→【管理】→【服务器设置】→【Samba 配置工具】，或在终端下键入 system-config-samba 命令。



图 3-1 Samba 服务器配置工具

3.1.1.1 配置服务器设置

配置 Samba 服务器的第一步是配置服务器的基本设置和几个安全选项。启动了应用程序后，选择【首选项】→【服务器设置】。【基本】活页标签如图 3-2 所示。



图 3-2 配置基本服务器设置

在【基本】标签上，指定计算机应在的【工作组】以及对计算机的简短的【描述】。它们与 smb.conf 中的 workgroup 和 server string 选项相对应。



图 3-3 配置安全服务器设置

【安全性】标签包含以下选项：

1) 【验证模式】

它和 security 选项相对应。选择以下验证模式中的一种。

a) 【ADS】

Samba 服务器充当活跃目录域（ADS）领域中的一个成员，Kerberos 在服务器上必须被安装和配置，并且 Samba 必须使用 net 工具成为 ADS 领域的一员。net 是 samba-client 软件包的一部分。详情请参考 samba-client 的说明书页。该选项不会把 Samba 配置成一个 ADS 控制器。

b) 【域】

Samba 服务器依赖于 Windows NT 主要或备份域控制器来校验用户。服务器把用户名和口令传递给控制器，然后等待它们被返回。在【验证服务器】字段中指定主要或备份域控制器的 NetBIOS 名称。如果【加密口令】选项被选，它必须被设置为【是】。

c) 【服务器】

Samba 服务器试图通过把用户名和口令组合传递给另一个 Samba 服务器来校验它们。如果它无法校验，服务器会试图使用用户验证模式来校验它们。

在**【验证服务器】**字段中指定另一个 Samba 服务器的 NetBIOS 名称。

d) **【共享】**

Samba 用户不必为每个 Samba 服务器都输入用户名和口令组合。它们在试图连接 Samba 服务器上的指定共享时才会被提示输入用户名和口令。

e) **【用户】**

(默认)Samba 用户必须为每个 Samba 服务器提供一个有效的用户名和口令。如果您想让在创建新用户时的 Windows 用户名选项生效, 选择这个选项。详情请参考下面的“3.1.1.2 管理 Samba 用户”。

2) **【加密口令】**

如果用户从 Windows 98、带有服务包 3 的 Windows NT 4.0、或其它最近版本的 Microsoft Windows 中连接, 该选项必须被启用。口令在服务器和客户间使用加密格式而非可被截取的纯文本格式传输。它和 encrypted passwords 选项相对应。关于加密 Samba 口令的详情, 请参考下面的“3.1.2 加密口令”。

3) **【来宾账号】**

当用户或来宾用户要登录入 Samba 服务器时, 他们必须被映射到服务器上的有效用户。选择系统上的现存用户名之一作为来宾 Samba 账号。当用户使用来宾账号登录 Samba 服务器, 他们拥有和这个用户相同的特权。该选项和 guest account 选项相对应。

点击了**【确定】**后, 所做改变会被写入配置文件, 守护进程会被重新启动, 因此改变会立即生效。

3.1.1.2 管理 Samba 用户

【Samba 服务器配置工具】要求在添加 Samba 用户之前, 在充当 Samba 服务器的系统上必须存在一个活跃的用户账号。Samba 用户和这个现存的用户账号相关联, 选择**【首选项】**→**【Samba 服务器设置】**。如图 3-4 所示。

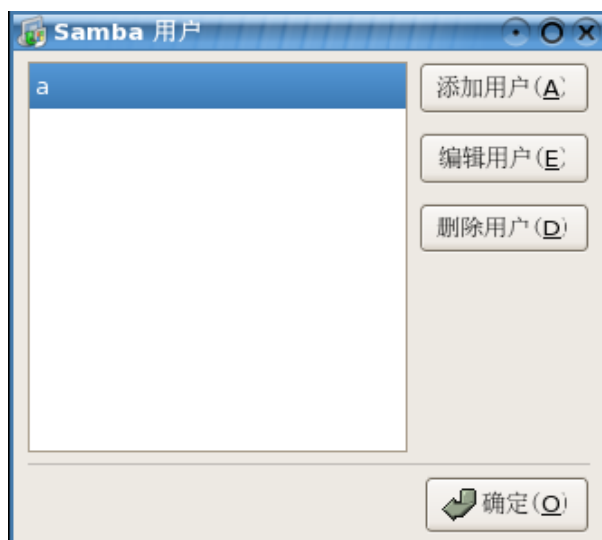


图 3-4 管理 Samba 用户

要添加 Samba 用户，选择【首选项】→【Samba 用户】，然后点击【添加用户】按钮。在【创建新 Samba 用户】窗口中的本地系统上的现存用户列表中选择【Unix 用户名】。



图 3-5 创建新 Samba 用户

如果用户在 Windows 机器上有一个不同的用户名，并将从 Windows 机器上登录 Samba 服务器，请在【Windows 用户名】字段中指定 Windows 用户名。【服务器设置】首选项的【安全性】活页上的【验证模式】必须被设置为【用户】才能使这个选项生效。

您还需要为 Samba 用户配置一个 Samba 口令，并再键入一次来确认这个口令。即便您选择了为 Samba 使用加密口令，仍建议您为所有用户设置一个不同于他们的系统口令的 Samba 口令。

要编辑某个现存用户，从列表中选择它，然后点击【**编辑用户**】。要删除某个现存的 Samba 用户，选择这个用户，然后点击【**删除用户**】按钮。删除 Samba 用户不会删除相关的用户账号。

点击了【**确定**】按钮后，用户就会被立即修改。

3.1.1.3 添加共享

选择【**文件**】→【**添加共享**】菜单，或直接点击工具栏里的【**添加共享**】按钮。如图 3-6，【**基本**】活页标签配置以下选项：

【**目录**】 — 需要通过 Samba 共享的目录，这个目录必须有所指定。

【**描述**】 — 对共享的简短描述。

【**基本权限**】 — 用户只能够读取共享目录中的文件或能够读写共享目录中的文件。

在【**访问**】活页标签上，选择是否要只允许指定的用户来访问共享还是允许所有 Samba 用户来访问共享。如果您选择了要允许指定用户访问，从可用的 Samba 用户列表中选择这些用户。

点击了【**确定**】按钮后，共享就会立即被添加。



图 3-6 创建 Samba 共享

3.1.2 加密口令

加密口令被默认启用，因为它更安全。如果加密口令没有被使用，纯文本口令就会被使用，它能够被别人使用网络分组嗅探器来截取。建议使用加密口令。

Microsoft SMB 协议最初使用纯文本口令。然而，带有服务包 3 或更高的

Windows NT 4.0、Windows 98、Windows 2000、Windows ME、以及 Windows XP 要求加密的 Samba 口令。要在 Linux 系统和运行以上 Windows 操作系统的系统间使用 Samba，您可以编辑 Windows 注册器来使用纯文本口令或配置您的 Linux 系统的 Samba 来使用加密口令。如果您要修改注册器，您必须为所有 Windows 机器这么做，这很冒险，有可能导致进一步的冲突。为了更高的安全性，推荐您使用加密口令。

要配置 Samba 使用加密口令，遵循以下步骤：

- 1) 为 Samba 创建一个单独的口令文件。要根据您的现存/etc/passwd 文件来创建，在 shell 提示下键入以下命令：

```
cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd
```

- 2) 如果系统使用 NIS，键入以下命令：

```
ypcat passwd | mksmbpasswd.sh > /etc/samba/smbpasswd
```

- 3) mksmbpasswd.sh 脚本和 samba 软件包一起安装在您的/usr/bin 目录上。

- 4) 改变 Samba 口令文件的权限许可，这样只有 root 用户才有读写权限：

```
chmod 600 /etc/samba/smbpasswd
```

- 5) 这个脚本不会把用户口令复制到新文件，Samba 用户账号在没有设置口令之前不会被激活。为了更高的安全性，建议您把用户的 Samba 口令设置为不同于用户的口令的口令。要设置每个 Samba 用户的口令，使用以下命令（把 username 替换为每个用户的用户名）：

```
smbpasswd username
```

- 6) 加密口令必须被启用。由于它们被默认启用，它们不必在配置文件中被特别启用。不过，它们也不能在配置文件中被禁用。在 smb.conf 文件中，请确定以下行不存在：

```
encrypt passwords = no
```

7) 如果它确实存在,请在行首加一个分号(;)来把它变成注释,这样该行就会被忽略,加密口令就会被启用。如果该行存在但没有被注释掉,请删除它或把它变成注释。

8) 要在配置文件中特别启用加密口令,给/etc/samba/smb.conf 文件添加以下几行:

```
encrypt passwords = yes  
smb passwd file = /etc/samba/smbpasswd
```

9) 在 shell 提示下键入 service smb restart 来确定 smb 服务被启动。

10) 如果您想让 smb 服务被自动启动,使用 ntsysv、chkconfig、或服务配置工具来在系统启动时自动启用它。

当使用了 passwd 命令后,pam_smbpass PAM 模块能够被用来同步用户的 Samba 口令和他们的系统口令。如果用户引发了 passwd 命令,他用来登录到中标麒麟服务器操作系统的口令以及他要连接 Samba 共享所必须提供的口令就会被改变。

要启用这个功能,把以下行添加到/etc/pam.d/system-auth 的 pam_cracklib.so 之下:

```
password required /lib/security/pam_smbpass.so nullok use_authok try_first_pass
```

3.1.3 启动和停止服务器

Samba 共享目录的服务器上必须运行 smb 服务。

使用以下命令来查看 Samba 守护进程的状态:

```
/sbin/service smb status
```

使用以下命令来启动守护进程:

```
/sbin/service smb start
```

使用以下命令来停止守护进程:

```
/sbin/service smb stop
```

要在引导时启动 smb 服务，使用以下命令：

```
/sbin/chkconfig --level 345 smb on
```

您还可以使用 `chkconfig`、`ntsysv` 或服务配置工具来配置在引导时启动哪些服务。

要查看到系统的活跃连接，执行 `smbstatus` 命令。

3.2 挂载共享

有时，您可能想把 Samba 共享挂载到目录上，这样该目录内的文件就如同是本地文件系统的一部分。

要把 Samba 共享挂载到某目录中，若该目录不存在则创建它，然后以 root 用户身份执行以下命令：

```
mount -t smbfs -o username=<username> //<servername>/<sharename> /mnt/point/
```

该命令会把<servername>中的<sharename>挂载在本地的/mnt/point/目录中。

3.3 手动配置 Samba

Samba 使用/etc/samba/smb.conf 作为它的配置文件。如果改变此配置文件，直到用 `smb` 命令服务重新启动 Samba 守护进程，此改变才生效。

中标麒麟高级服务器操作系统中的默认配置文件 `smb.conf` 允许用户在以同样的用户名和口令注册后浏览 Windows 机器上作为 Samba 共享的 Linux 主目录。还可作为 Samba 共享打印机共享为中标麒麟高级服务器操作系统配置的所有打印机，换句话说，可将打印机连到中标麒麟高级服务器操作系统并从网络上装有 Windows 系统的机器上进行打印。

为了指定 Windows 工作组和描述串，编辑文件 `smb.conf` 中的下列行：

```
workgroup = WORKGROUPNAME
server string = BRIEF COMMENT ABOUT SERVER
```

用这台机器应属于的 Windows 工作组的名字取代 `WORKGROUPNAME`。
`BRIEF COMMENT ABOUT SERVER` 是可选的，是关于 Samba 系统的 Windows 注释。

要在 Linux 系统上创建 Samba 共享目录，将以下部分加入到 `smb.conf` 文件中（进行修改以反映用户的需要）：

```
[sharename]
comment = Insert a comment here
path = /home/share/
valid users = tfox carole
public = no
writable = yes
printable = no
create mask = 0765
```

上述例子允许用户 `tfox` 和 `carole`，在 Samba 服务器上，从 Samba 客户机认证后登录到目录 `/home/share`。

3.4 连接到 Samba 共享

为了从 Microsoft Windows 机器连接到 Linux Samba 共享，使用网上邻居或 Windows 资源管理器。

要从 Linux 系统连接到 Samba 共享，键入下列命令：

```
smbclient //hostname/sharename -U username
```

需用连接到 Samba 服务器的主机名或 IP 地址替换 `hostname`，用要浏览的共享目录名替换 `sharename`，用该系统的 Samba 用户名替换 `username`。如果不需要给用户口令，输入当前口令或按[Enter]键。

如果看到 `smb:\>` 提示符，表明注册成功了。要看命令列表键入 `help`。如果希望浏览主目录的内容，用用户名替换 `sharename`。如果不需要 `-U` 开关，则当前用户的用户名被传递给 Samba 服务器。

要退出 `smbclient`，在 `smb:\>` 提示符下键入 `exit`。

3.5 如何在中标麒麟高级服务器操作系统下使用加密口令配置 Samba

最初微软 SMB 协议使用明文口令。而 Windows 2000 和 Windows NT 4.0 用服务包 3 或更高要求的加密 Samba 口令，要在中标麒麟高级服务器操作系统和 Windows 2000 或 Windows NT 4.0 服务包 3 或更高系统之间使用 Samba，可使用明文口令编辑 Windows registry 或在 Linux 系统上使用加密口令配置 Samba。如

果选择编辑 **registry**，则必须在到目前为止的所有 Windows NT 或 2000 机器上进行，这是有风险的且可能产生进一步冲突。

要在中标麒麟高级服务器操作系统上使用加密口令配置 Samba，按下列各步进行：

- 1) 为 Samba 创建一个单独的口令文件。要在现有的/etc/passwd 文件的基础上创建，在 shell 提示符下键入下列命令：

```
cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd
```

- 2) mksmbpasswd.sh 脚本和 samba 包一起安装在目录/usr/bin 下。
- 3) 使用命令 `chmod 600 /etc/samba/smbpasswd` 改变 Samba 口令文件的权限，因为只有根有读写权限。
- 4) 脚本不复制用户口令给新文件。要设置每个 Samba 用户口令使用命令 `smbpasswd username`（用各个用户的用户名替换 `username`）。直到为 Samba 用户设置了 Samba 口令，Samba 用户帐号才被激活。
- 5) Samba 配置文件中对口令进行加密。在文件 `smb.conf` 中，把下面行的注释去掉：

```
encrypt password = yes
smb passwd file = /etc/samba/smbpasswd
```

- 6) 为使改变生效，在 shell 提示符下输入命令 `service smb restart` 重新启动 Samba。

要阅读更多关于使用 Windows NT 4.0 和 Windows 2000 的 Samba，阅读目录 `/usr/share/doc/samba-version-number/docs/textdocs/` 中的 `ENCRYPTION.txt`、`Win95.txt` 和 `WinNT.txt`（已安装的 Samba 的版本号代替 `version-number`）。

3.6 附加资源

对于未包括在这里的配置文件，可查阅下面的资源。

`smb.conf` 的说明书页（man）— 解释该如何配置 Samba 配置文件。

`smbd` 的说明书页（man）— 描述 Samba 守护进程的工作原理。

`smbclient` 和 `findsmb` 的说明书页 — 进一步学习这些客户工具。

`/usr/share/doc/samba-<version-number>/docs/` — 包括在 samba 软件包中的帮

助文件。

<http://www.samba.org> — Samba 网页，包含许多关于邮件列表和 GUI 界面列表信息等的有用文档。

http://www.samba.org/samba/docs/using_samba/toc.html — 由 Jay Ts、Robert Eckstein 和 David Collier-Brown 编纂，O'Reilly & Associates 出版的 Using Samba, 2nd Edition 一书的在线版本。

4 NFS 配置和使用

NFS（Network File System，网络文件系统）是对网上不同机器之间共享文件的一种方法，就像不同机器上的文件被放在本地硬盘上一样。中标麒麟高级服务器操作系统既是 NFS 服务器也是 NFS 客户，这表明既可向其它机器输出文件系统，也可安装其他机器输出的文件系统。

4.1 为什么使用 NFS？

NFS 对相同网络上多用户之间文件的共享目录是有用的。例如，在同一个项目上工作的一组用户有权使用文件，即项目使用安装在目录/myproject 里的 NFS 文件系统共享部分的文件（NFS 共享），为访问共享文件，用户进入那台机器上的/myproject 目录不需要输入口令或记住专用命令，用户好像工作在自己的本地机器目录上。

4.2 安装 NFS 文件系统

使用 mount 命令从其它机器安装 NFS 文件系统：

```
mount shadowman: /mnt/export /mnt/local
```

本地机器的安装点目录必须是已经存在的（上例中的/mnt/local）。

在此命令中，shadowman 是 NFS 文件服务器的主机名，/mnt/export 是 shadowman 正在输出的文件系统，/mnt/local 是存放要安装文件系统在本地图器上的目录。mount 命令运行后（如果 shadowman 有适当的权限），我们可输入 ls /mnt/local，获得 shadowman 上目录/mnt/export 中文件的列表。

4.2.1 使用/etc/fstab 安装 NFS 文件系统

从另一台机器安装共享 NFS 的方法是加一行到文件/etc/fstab 中。此行必须规定 NFS 服务器的主机名、服务器上输出的目录、安装文件系统的本地机器上的目录。只有 root 用户能修改文件/etc/fstab。

/etc/fstab 中行的一般语法是：

```
server: /usr/local/pub /pub nfs rsize=8192,wsizer=8192,timeo=14,intr
```

机器上的安装点/pub 必须存在。将这一行加到/etc/fstab 中后, 可在 shell 提示符下输入命令 mount/pub, 从服务器安装/pub。

4.2.2 使用 autofs 安装 NFS 文件系统

安装 NFS 共享的第三个选项是 autofs 的使用。Autofs 用 automount 守护进程由仅在访问它们时动态安装来管理安装点。

autofs 咨询主映射配置文件/etc/auto.master 来决定要定义哪些挂载点。然后, 它使用适用于各个挂载点的参数来启动 automount 进程。主映射配置中的每一行都定义一个挂载点, 一个分开的映射文件定义在该挂载点下要挂载的文件系统。譬如, /etc/auto.misc 文件可能会定义/misc 目录中的挂载点; 这种关系在 /etc/auto.master 文件中会被定义。

auto.master 中的每项都有三个字段。第一个字段是安装点, 第二个字段是映射文件的位置, 第三个字段是可选的。第三个字段包含如暂停值的信息。

例如, 为了将远程机器 penguin.host.net 上的目录/project52 安装在本机的安装点/mnt/myproject, 将下面的行加到 auto.master 中:

```
/mnt /etc/auto.mnt --timeout 60
```

将下列行加到文件/etc/auto.mnt 中:

```
myproject -rw,soft,intr,rsiz=8192,wsiz=8192 penguin.host.net: /project52
```

/etc/auto.mnt 中第一个字段是/mnt 子目录的名字, automount 动态创建本目录, 实际它不在客户机器上。第二个字段包括 mount 选项, 如用于读写访问的 rw。第三个字段是 NFS 输出的位置, 包括主机名和目录。

目录/mnt 必须在本地文件系统上。本地文件系统上的/mnt 没有子目录。

Autofs 是服务。要启动此服务, 在 shell 提示符下输入下面的命令:

```
service autofs restart
```

为浏览活动安装点, 在 shell 提示符下输入下面的命令:

```
service autofs status
```

如果在 `autofs` 运行时修改 `/etc/auto.master` 配置文件，则必须告诉 `automount` 守护进程重新下载。在 `shell` 提示符下输入下面的命令：

```
service autofs reload
```

使用 `TCP` 作为传输协议。

`NFS` 的默认传输协议是 `UDP`；然而，现在的 `linux` 内核提供了对通过 `TCP` 的 `NFS` 的支持。要通过 `TCP` 来使用 `NFS`，在客户系统上安装 `NFS` 导出的文件系统时，包括一个 `-o tcp` 选项。例如：

```
mount -o tcp shadowman.example.com: /misc/export /misc/local
```

如果 `NFS` 在 `/etc/fstab` 中被指定：

```
server: /usr/local/pub /pub nfs rsize=8192,wsiz=8192,timeo=14,intr,tcp
```

如果它在 `autofs` 配置文件中被指定：

```
myproject -rw,soft,intr,rsize=8192,wsiz=8192,tcp penguin.example.net: /project52
```

由于默认协议是 `UDP`，如果没有指定 `-o tcp` 选项，`NFS` 导出的文件系统就会通过 `UDP` 来进入。

使用 `TCP` 的优越性包括：

- 1) 提高了的连接持久性，因此获得的 `NFS stale file handles` 消息就会较少。
- 2) 负载量较大的网络的性能会有所提高。因为 `TCP` 确认每个分组，不像 `UDP` 只在完成时才确认。
- 3) `TCP` 的拥塞控制技术比 `UDP` 要好（`UDP` 根本没有）。在一个拥塞情况严重的网络上，`UDP` 分组是被首先撤消的类型。这意味着，如果 `NFS` 正在写入数据（单元为 `8K` 的块），所有这 `8K` 数据都需要被重新传输。由于 `TCP` 的可靠性，`8K` 中只有一部分需要重新传输。
- 4) 错误检测。当 `tcp` 连接中断（由于服务器停运），客户就会停止发送数据而开始进行重新连接。`UDP` 是无连接的，使用它的客户就会继续发送网络数据直到服务器重新上线为止。

主要的不利因素是，由于 TCP 协议的费用，在性能方面的提高并不显著。

4.2.3 ACL 支持

ACL 即访问存取控制列表（Access Control Lists, ACL）。Linux 内核为 ext3 文件系统和使用 NFS 或 Samba 协议安装的 ext3 文件系统提供了 ACL 支持。这样，如果某个 ext3 文件系统启用了 ACL，而且被 NFS 导出了，如果 NFS 客户能够读取 ACL，它们就能够被 NFS 客户使用。

按照默认设置，如果被 NFS 服务器导出的文件系统支持 ACL，并且 NFS 客户能够读取 ACL，ACL 就会被客户系统利用。

在配置服务器的时候，若要禁用 NFS 共享上的 ACL，则在/etc/exports 文件中加入 no_acl 选项。要在客户端上安装 NFS 共享的时候禁用其中的 ACL，通过命令行或/etc/fstab 文件使用 no_acl 选项来安装它。

4.3 导出 NFS 文件系统

从 NFS 服务器中共享文件又称导出目录。NFS 服务器配置可以用来把系统配置成 NFS 服务器。

4.3.1 图形化配置工具

要使用 NFS 服务器配置工具，您必须具备 root 权限。要启动这个程序，点击面板上的【启动】→【系统】→【管理】→【服务器配置】→【NFS 服务配置】，或键入 system-config-nfs 命令。



图 4-1 NFS 服务器配置工具

要添加 NFS 共享，点击【添加】按钮。如图 4-1 所示的对话框会出现。

【基本】 活页标签要求以下信息：

【目录】 — 指定要共享的目录，如/tmp。

【主机】 — 指定要共享目录的主机。请参考下面的“4.3.3 主机名格式”来获取对格式的解释。

【基本权限】 — 指定目录应该有只读权限还是读写权限。

【一般选项】 活页标签（如图 4-2）允许您配置以下选项：

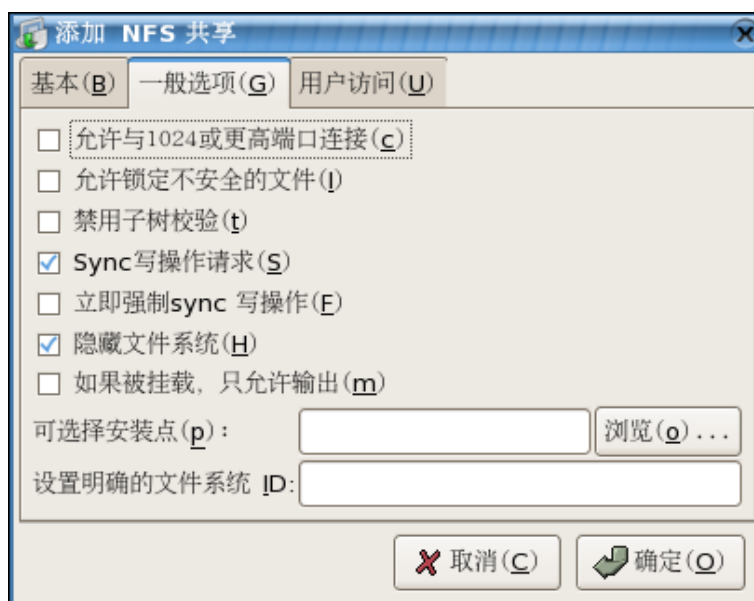


图 4-2 一般选项

1) 【允许与 1024 或更高端口连接】

在号码小于 1024 的端口上启动的服务必须以 root 用户身份启动。选择这个选项来允许 root 用户以外的用户来启动 NFS 服务。该选项和 insecure 相对应。

2) 【允许锁定不安全的文件】

不需要锁定请求。该选项和 insecure_locks 相对应。

3) 【禁用子树检查】

如果某文件系统的子目录被导出，但是整个文件系统没有被导出，服务器会检查所请求的文件是否在导出的子目录中。这种检查叫做子树检查（subtree checking）。选择这个选项来禁用子树检查。如果整个文件系统被导出，选择禁用子树检查可以提高传输率。该选项和 no_subtree_check 相对应。

4) 【Sync 写操作请求】

默认被启用，该选项不允许服务器在请求被写入磁盘前回复这些请求。该选项和 sync 相对应。如果它没有被选择，async 选项会被使用。

5) 【立即强制 sync 写操作】

不推迟写入磁盘的操作。该选项和 no_wdelay 相对应。

【用户访问】活页标签（如图 4-3）允许您配置以下选项：



图 4-3 用户访问

1) 【将远程根目录用户视为本地根目录用户】

按照默认设置，root 用户的用户 ID 和组群 ID 都是 0。root 权限压缩（root squashing）把用户 ID 0 和组群 ID 0 映射为匿名的用户和组群 ID，因此客户上的 root 用户就不会在 NFS 服务器上具备 root 权限。如果这个选项被选，root 用户就不会被映射为匿名用户，客户上的 root 用户就会对导出的目录拥有 root 权限。选择这个选项会大大降低系统的安全性。除非绝对必要，请不要选择它。该选项和 no_root_squash 相对应。

2) 【将所有客户端用户视为匿名用户】

如果该选项被选，所有用户和组群 ID 都会被映射为匿名用户。该选项和 all_squash 相对应。

3) 【为匿名用户指定本地用户 ID】

如果将所有客户端用户视为匿名用户，这个选项会让您为匿名用户指定一个用户 ID。该选项和 corresponds to anonuid 相对应。

4) 【为匿名用户指定本地组 ID】

如果将所有客户端用户视为匿名用户，这个选项会让您为匿名用户指定一个组群 ID。该选项和 corresponds to anongid 相对应。

【NFS 服务器配置工具】直接读写/etc/exports 配置文件。因此，这个文件在使用该工具后可以被手工修改；手工修改了该文件后也可以使用这个工具（假

定手工修改时使用了正确的语法)。

4.3.2 命令行配置

如果您更喜欢使用文本编辑器来编辑配置文件或者没有安装 X 窗口系统，可以直接修改配置文件。

/etc/exports 文件控制 NFS 服务器要导出哪些目录。它的格式如下：

```
directory hostname(options)
```

唯一需要指定的选项是 sync 和 async 之一(建议使用 sync)。如果指定了 sync，服务器在请求所做的改变被写入磁盘之前就不会回复这些请求。

例如：

```
/misc/export speedy.example.com(sync)
```

会允许来自 speedy.example.com 的用户使用默认的只读权限来挂载 /misc/export，但是：

```
/misc/export speedy.example.com(rw, sync)
```

将会允许来自 speedy.example.com 的用户使用读写权限来挂载 /misc/export。

请参考下面的小节来获取关于主机名格式的解释。

请谨慎处理/etc/exports 文件中的空格。如果主机名和括号内的选项之间没有空格，这些选项就只应用于这个主机名。如果在主机名和选项之间有空格，这些选项就是全局应用的。例如，请仔细查看以下行：

```
/misc/export speedy.example.com(rw, sync)  
/misc/export speedy.example.com (rw, sync)
```

第一行给来自 speedy.example.com 的用户以读写权限并拒绝所有其他用户。第二行给来自 speedy.example.com 的用户以只读权限（默认设置），并给予所有其他用户以读写权限。

在您每次改变/etc/exports 的时候，您必须把改变通知给 NFS 守护进程，或使用以下命令来重新载入配置文件：

```
/sbin/service nfs reload
```

4.3.3 主机名格式

主机可以使用以下格式：

1) 单个机器

一个全限定域名（能够被服务器解析的），主机名（能够被服务器解析的），或 IP 地址。

2) 使用通配符来指定的机器系列

使用 * 或 ? 字符来指定一个字符串匹配。IP 地址中不使用通配符；不过如果反向 DNS 查询失败，它们可能会碰巧有用。在完整域名中指定通配符时，点 (.) 不包括在通配符中。例如：*.example.com 包括 one.example.com，但不包括 one.two.example.com.。

3) 网络

使用 a.b.c.d/z，这里的 a.b.c.d 是网络，z 是子网掩码中的位数（如 192.168.0.0/24）。另一种可以接受的格式是 a.b.c.d/netmask，这里的 a.b.c.d 是网络，netmask 是子网掩码（如 192.168.100.8/255.255.255.0）。

4) Netgroups

格式为@group-name，这里的 group-name 是 NIS netgroup 的名称。

4.3.4 启动和停止服务器

在导出 NFS 文件系统的服务器上，nfs 服务必须在运行。

使用以下命令来查看 NFS 守护进程的状态：

```
/sbin/service nfs status
```

使用以下命令来启动 NFS 守护进程：

```
/sbin/service nfs start
```

使用以下命令来停止 NFS 守护进程：

```
/sbin/service nfs stop
```

要在引导时启动 NFS 服务，使用以下命令：

```
/sbin/chkconfig --level 345 nfs on
```

您还可以使用 `chkconfig`、`ntsysv` 或服务配置工具来配置在引导时启动哪些服务。

4.4 附加资源

本章讨论使用 NFS 的基础知识。关于更详细的信息，参考下面的资源。

`nfsd` (8)、`mountd` (8)、`exports` (5)、`auto.master` (5)、`autofs` (5) 和 `autofs` (8) 联机手册 — 这些联机手册给出 NFS 和 `autofs` 配置文件的正确语法。

<http://nfs.sourceforge.net/> — NFS 网站，包括到邮件列表和 FAQ 的链接。

<http://www.tldp.org/HOWTO/NFS-HOWTO/index.html> — 来自 Linux 文档计划的 Linux NFS-HOWTO。

Hal Stern 写的《管理 NFS 和 NIS 服务》(Managing NFS and NIS Services)，O'Reilly & Associates, Inc. 出版。

5 邮件服务器配置和使用

电子邮件始于 20 世纪 60 年代。邮箱是用户主目录里的一个文件，只有其用户可读。早期的邮件应用程序在文件底部追加新的文本信息，使文件不断增大，用户要越过这些附加的信息才能找到所需的信息。这种系统只能给同系统内的用户发送信息。

1971 年，人们第一次在网络上使用电子邮件，当时一名叫雷·汤姆林计算机工程师在两台机器之间通过 ARPANET（因特网的前身）发送测试信息。电子邮件通讯方式很快就流行起来，不到两年，这种通信就占了 ARPANET 通信量的 75%。

于今，基于标准化网络协议的邮件系统演变成最广泛使用的互联网服务之一。中标麒麟高级服务器操作系统给出了许多先进的应用程序来为电子邮件提供服务和访问。

本章回顾了目前使用的现代电子邮件协议，有些程序设计为发送和接收邮件。

5.1 电子邮件协议

今天，人们使用客户端/服务器架构来传送电子邮件。使用邮件客户端程序创建电子邮件消息。然后此程序会向服务器发送消息。然后，此服务器把消息转发给收件人的电子邮件服务器，接下来会将此消息提供给收件人的电子邮件客户端。

要完成这个过程，各类标准网络协议允许运行不同操作系统和使用不同邮件程序的不同设备来发送和接收邮件。

以下提到的是最常用的邮件传输协议。

5.1.1 邮件传输协议

邮件从客户端发送到服务器端，以及从源服务器发送到目标服务器，都是由简单邮件传输协议（SMTP）来处理的。

5.1.1.1 SMTP

SMTP 的主要用途是在邮件服务器之间传递邮件。但是，SMTP 对于邮件客户端也很重要。为了发送电子邮件，客户端需要发送消息到外寄邮件服务器，外

寄邮件服务器随后联系目标邮件服务器，从而完成邮件传输。因此，在配置邮件客户端时，有必要明确规定一个 SMTP 服务器。

中标麒麟高级服务器操作系统下，用户可以配置本地机器上的 SMTP 服务器来处理邮件的投递。然而，也可配置远程 SMTP 服务器来发送邮件。

使用 SMTP 协议很重要的一点是它不要求身份认证。使用 SMTP 协议时，不必通过身份认证，网络上的任何人就可以向其他人、甚至向一大群人发送电子邮件。正是这种特点导致了大量垃圾邮件。强加中继约束来限制互联网上任意用户通过你的 SMTP 服务器向互联网上其他服务器发送电子邮件。不强加这种约束的服务器被称为开放中继服务器。

中标麒麟高级服务器操作系统提供了 Postfix 软件和 Sendmail SMTP 软件。

5.1.2 邮件存取协议

邮件客户端应用程序使用以下两个主要协议完成从邮件服务器检索电子邮件的工作：

邮局协议(POP)和因特网消息访问协议(IMAP)。

5.1.2.1 POP

中标麒麟高级服务器操作系统默认的 POP 服务器是 Dovecot，由 dovecot 软件包提供。

安装 dovecot 包：

要使用 Dovecot，请首先确保您的系统已安装 Dovecot 包，要安装此包，请作为 root 用户运行以下命令：

```
yum install dovecot
```

使用 POP 服务器时，由邮件客户端应用程序下载邮件消息。默认情况下，大多 POP 电子邮件客户端都被自动配置为：邮件传输成功后删除邮件服务器上的消息，然而通常也可改变此设置。

POP 与重要的互联网信息标准完全兼容，如 MIME（Multipurpose Internet Mail Extensions，多功能 Internet 邮件扩充服务），此标准允许为电子邮件增加附件。

对于具备电子邮件阅读系统的用户来说，POP 能发挥最好的作用。对于不

能持续连接到因特网或邮件服务器所在网络的那些用户来说，POP 也能起到很好的作用。不幸的是，对于那些网络连接慢的用户来说，POP 需要基于身份认证的客户端程序才能下载每个消息的所有内容。如果消息含有较大的附件，这需要花很长时间。

POP3 是目前使用最多的标准 POP 协议版本。

也有很多使用较少的 POP 协议变体：

1) APOP

附带 MDS 认证的 POP3 协议。从邮件客户端发送到服务器的是用户密码的哈希码而不是未加密的密码。

2) KPOP

附带 Kerberos 认证的 POP3 协议。它使用特定于用户的一个 ID（类似于密码）来认证 POP 请求。然而，这个 ID 并不是加密的，所以 KPOP 并不比标准 POP 更安全。

为增强安全性，可使用 Secure Socket Layer(SSL)对客户端认证和数据传递对话进行加密。这可以通过使用 POP3 服务或通过使用/usr/sbin/stunnel 应用程序来实现。更多关于加强邮件通讯安全的信息，参考 5.5.1 安全通讯。

5.1.2.2 IMAP

中标麒麟高级服务器操作系统默认的 IMAP 服务器是 Dovecot，由 dovecot 软件包提供。关于安装 Dovecot 的详细信息，请参考 5.1.2.1 POP。

当使用 IMAP 邮件服务器时，邮件消息保留在服务器上，用户可以阅读或删除此服务器上的电子邮件。IMAP 也允许客户端应用创建、重命名或删除服务器上的邮件目录并整理和存储电子邮件。

对于使用多个机器访问电子邮件的用户来说，IMAP 尤其有用。对于以很慢的速度连接到邮件服务器的用户来说，IMAP 也很方便，因为在打开邮件前，只下载电子邮件标题信息，这样做节省了带宽。用户也可以直接删除邮件，而不查看或者不下载邮件。

为方便起见，IMAP 客户端应用程序可以在本地缓存消息副本，因此用户可以在没有直接连接到 IMAP 服务器时，浏览以前读过的消息。

如 POP 一样，IMAP 可以与重要的互联网信息标准完全兼容，如 MIME，

MIME 标准允许为电子邮件增加附件。

为增强安全性,可使用 Secure Socket Layer(SSL)对客户认证和数据传递对话加密。这可以通过 imap 服务或通过使用/usr/sbin/stunnel 程序来实现。更多关于加强邮件通讯安全的信息,请参考 5.5.1 安全通讯。

也有其他免费和收费的 IMAP 客户端和服务端,其中许多 IMAP 客户端和服务端扩展了 IMAP 协议并提供额外的功能。

5.1.2.3 Dovecot

使用 IMAP 和 POP3 协议的 imap-login 和 pop3-login 进程来自于 dovecot 包中的主 dovecot 守护进程。用户可以通过/etc/dovecot/dovecot.conf 配置文件来设置 IMAP 和 POP 的用法,默认情况下, Dovecot 同时运行使用 SSL 的 IMAP 安全版和 POP3 安全版。若要通过配置 dovecot 来使用 POP, 请完成下列步骤:

- 1) 编辑/etc/dovecot/dovecot.conf 配置文件。

确保没有取消对 protocols 变量的注释(删除这行起始的(#)标志), 并且 protocols 变量包括 POP3 参数。例如:

```
protocols = imap imaps pop3 pop3s
```

当取消对 protocols 变量的注释时, dovecot 将使用这个变量的默认值。

- 2) 通过运行下列命令, 使配置变更对当前会话生效:

```
service dovecot restart
```

- 3) 通过运行下列命令, 使配置变更在下次重启后生效:

```
chkconfig dovecot on
```

- 4) dovecot 服务启动 POP3 服务器。

请注意, dovecot 只报告它开启了 IMAP 服务器,但实际上它也启动了 POP3 服务器。

和 SMTP 不同, IMAP 和 POP3 都需要使用用户名和密码对连接的客户进行认证。默认情况, 两种协议的密码通过未加密形式传递到网络。

为配置 dovecot 上的 SSL, 请完成下列步骤:

- 1) 根据您的需要，编辑/etc/pki/dovecot/dovecot-openssl.conf 配置文件。但是，典型安装中，这个文件不需要修改。
- 2) 重命名，移动或者删除/etc/pki/dovecot/certs/dovecot.pem 和/etc/pki/dovecot/private/dovecot.pem 文件。
- 3) 执行/usr/libexec/dovecot/mkcert.sh 脚本，该脚本创建了 dovecot 自签名证书。这些证书被复制在/etc/pki/dovecot/certs 和/etc/pki/dovecot/private 目录中。为使上述修改生效，请重新启动 dovecot:

```
service dovecot restart
```

更多关于 dovecot 信息，参考网站 <http://www.dovecot.org>。

5.2 电子邮件程序分类

总的来说，所有电子邮件应用程序分为至少 3 类。每类在移动和管理邮件消息过程中都起特定的作用。大部分用户只知道他们是用来接收和发送邮件的特定程序，每种程序对于保证邮件能正确到达目的地都起重要作用。

5.2.1 邮件传输代理

MTA (Mail Transfer Agent, 邮件传输代理) 通过 SMTP 在不同主机间传送电子邮件消息。消息在传送到目标地前，它涉及了几种 MTA。

虽然不同机器间的消息投递看起来是直接转发的，但是决定某个 MTA 是否可以或者应该接收要投递的消息的整个过程是相当复杂的。另外，由于垃圾邮件的问题，MTA 通常受到 MTA 配置或 MTA 所在网络访问配置的限制。

发送邮件时，很多现代电子邮件客户端程序可以起到与 MTA 一样的作用。但是，这个作用不能与 MTA 的真正作用混淆。电子邮件客户端程序可以像 MTA 一样发送邮件的唯一原因是，运行应用程序的主机没有自己的 MTA。在非基于 UNIX 操作系统上，电子邮件客户端程序就是这样。然而，这些客户端程序只是把约束的信息发送到授权使用的 MTA 上，而不是直接发送到目标接收者的电子邮件服务器上。

自从中标麒麟高级服务器操作系统提供了两个 MTA—Postfix 和 Sendmail，就不再需要把电子邮件客户端程序作为 MTA 使用。中标麒麟高级服务器操作系统还包含特殊用途的 MTA，叫做邮件工具 (Fetchmail)。

5.2.2 邮件投递代理

MDA (Mail Delivery Agent, 邮件投递代理) 被 MTA 调用, 把即将收到的邮件分到合适的用户邮箱。很多情况下, MDA 实际是本地投递代理 (LDA), 例如邮件或邮件工具。

可以把消息投递到电子邮件客户端的程序, 可阅读位置的任何程序, 都可以称之为 MDA。正因为这个原因, 当一些 MTA (例如 Sendmail 和 Postfix) 将新邮件消息附加到本地用户的邮件假脱机文件中时, 它们都可以充当 MDA 的角色。总的来说, MDA 不在系统间传送消息, 也不为用户提供接口。MDA 只是在本地机器上分发和分类消息, 从而使电子邮件用户应用程序可以访问这些消息。

5.2.3 电子邮件用户代理

MUA (Mail User Agent, 邮件用户代理) 和电子邮件客户端应用程序同义。MUA 是让用户可以读取和编辑邮件消息的程序。很多 MUA 程序都可以通过 POP 或者 IMAP 协议检索消息, 设置邮箱以储存消息和发送约定消息到 MTA。

MUA 也可以是图形化的, 例如 Thunderbird, 或者有简单的文本接口, 例如 pine。

5.3 邮件传输代理

中标麒麟高级服务器操作系统提供两种主要的 MTA—Postfix 和 Sendmail。Postfix 为 MTA 的默认设置, 尽管从默认设置调到 sendmail 很容易。要从默认 MTA 调到 sendmail, 你可以通过卸载 Postfix 或者使用以下命令:

```
alternatives --config mta
```

你还可以通过以下命令来激活/取消需要的服务:

```
chkconfig <service> <on/off>
```

5.3.1 Postfix

最初是由 IBM 安全专家和程序员 Wietse Venema 研发出的, Postfix 是与 Sendmail 兼容的 MTA, 它是为实现安全, 快速, 简单配置而设计的。

为增强安全性, Postfix 使用模块设计。其中, 主守护进程启动权限有限、

进程小的、权限低的进程，只能执行与邮件投递各个阶段相关的具体任务，并运行在一种固定变化的环境中，从而限制攻击照成的影响。

要把 Postfix 配置成接收来自于主机而不是本地电脑的网络连接，只须在其配置文件里做很小的改变。至于那些更复杂的需求，Postfix 提供多种配置选项和第三方插件，使其成为通用，全面的 MTA。

Postfix 的配置文件是可读的文件，可支持最多 250 条指令。不像 Sendmail、Postfix 不需要宏进程来使配置变更生效，多数最常用的选项在含有大量注释的文件中都有所描述。

5.3.1.1 默认 Postfix 安装

Postfix 可执行文件是/usr/sbin/postfix。此守护进程启动处理邮件邮递的所有相关过程。

Postfix 把其配置文件储存在/etc/postfix/目录下。以下是常用文件列表：

1) Access

用于访问控制，该文件指定允许哪些主机连接到 Postfix。

2) main.cf

全局配置文件。大多数配置选项在此文件中指明。

3) master.cf

指明 Postfix 如何和各种进程相互作用以完成邮件投递。

4) transport

地图电子邮件地址传递主机。

在/etc/里可以找到 aliases 文件。Postfix 和 Sendmail 共享这个文件。此文件是描述用户 ID 别名的邮件协议所需要的可配置列表。

为其他用户配置 Postfix 成一个服务器：

默认的/etc/postfix/main.cf 文件不允许 Postfix 接收除了本地电脑以外的连接。为其他用户配置 Postfix 成服务器的具体说明，参考 5.3.1.2 基本 Postfix 配置。

修改/etc/postfix 目录下任何配置文件，需重启 postfix 服务器，否则修改不能生效：

```
service dovecot restart
```

5.3.1.2 基本 Postfix 配置

默认情况下，Postfix 不接收除了本地电脑以外的连接。请以 root 用户执行以下步骤，使网络中其他主机也可以投递电子邮件。

- 1) 用文本编辑器（如 vi）来编辑/etc/postfix/main.cf 文件。
- 2) 通过删除（#）符号来取消对 mydomain 行的注释，并用邮件服务器服务的域名替代 domain.tld，如 example.com。
- 3) 取消对 myorigin=\$mydomain 行的注释。
- 4) 取消对 myhostname 行的注释，并用主机名替换 host.domain.tld。
- 5) 取消对 mydestination=\$myhostname, localhost.\$mydomain 行的注释。
- 6) 取消对 mynetworks 行的注释，用连接到服务器的有效网络设置替代 168.100.189.0/28。
- 7) 取消对 inet_interfaces=all 行的注释。
- 8) 增加对 inet_interfaces=localhost 行的注释。
- 9) 重启 postfix 服务器。

完成这些步骤后，主机即可以接收外面投递的邮件。

Postfix 有多类配置选项。学习配置 Postfix 的最好方法是阅读 etc/postfix/main.cf 配置文件中的注释。包括 Postfix 配置、SpamAssassin 集成、及/etc/postfix/main.cf 参数详细描述的其他资源，可以在 <http://www.postfix.org/>找到。

5.3.1.3 利用 LDAP 使用 Postfix

Postfix 可以使用 LDAP 目录作为不同查找表(如:aliases、virtual、canonical 等)的来源。这使 LDAP 可以存储分层用户信息，并使得只有 Postfix 能在必要时得到 LDAP 查询的结果。通过本次存储的信息，管理员可以很容易地维护它。

5.3.1.3.1 /etc/aliases 查找表示例

以下示例是使用 LDAP 查找/etc/aliases 文件。确保 etc/postfix/main.cf 包含下信息：

```
alias_maps=hash:/etc/aliases, ldap:/etc/postfix/ldap-aliases.cf
```

如果没有已创建的/etc/postfix/ldap-aliases.cf 文件，请创建一个新的，并确保

它包含以下信息：

```
server_host=ldap.example.com
search_base=dc=example,dc=com
```

ldap.example.com、example 和 com 这些参数都需被 LDAP 服务器上的参数替代。

/etc/postfix/ldap-aliases.cf 文件可以指定各种参数，包括启用 LDAP SSL 和 STARTTLS 的参数。

5.3.2 Sendmail

Sendmail 的主要作用和 MTA 一样，在主机之间安全传送邮件，通常使用 SMTP 协议。然而，Sendmail 是高度可配置的，允许控制几乎邮件处理的每个方面，包括使用的协议。由于 Sendmail 的强大和可扩展性，许多系统管理员都把 Sendmail 选作 MTA。

5.3.2.1 目的和局限性

我们有必要了解什么是 Sendmail 以及它的功能。在流行多角色大型应用的今天，sendmail 看上去是运行一个组织的邮件服务器所必需的的唯一应用程序。从技术上讲，这是事实，因为 sendmail 可以把邮件分发到用户目录以及为用户发送邮件。然而，大多数用户的实际需要远远不止于简单的邮件投递。用户通常想用 MUA 来与电子邮件互动，用 POP 或者 IMAP 把信息下载到他们的本地设备。或者，他们可能更喜欢使用 Web 界面来访问他们的邮箱。这些其他的应用程序可以和 sendmail 同时工作，他们的存在也确实有多种原因，这些应用程序可以单独运行。

配置 Sendmail 的方法超出了本节的范围。关于 Sendmail 的功能以及如何在错误时进行故障修复的信息已被收集成卷，其中也包含了许许多多的选项和规则集信息。关于 Sendmail 资源的列表，请参考 5.6 更多信息资源。

下一部章节讲述默认与 sendmai 一起安装的文件，并回顾基本的配置变更，包括如何阻止不必要的电子邮件(垃圾邮件)和如何用轻量级目录访问协议(LDAP)来扩展 Sendmail。

5.3.2.2 默认 Sendmail 安装

为使用 Sendmail，请首先确保 Sendmail 包已安装在您的系统中，要做到这

一点，作为 root 用户运行：

```
yum install dovecot
```

更多关于如何切换默认 MTA 的信息，请参考 5.3 邮件传输代理。

Sendmail 的可执行文件是/usr/sbin/sendmail。

Sendmail 的冗长和详细配置文件是/etc/mail/sendmail.cf。请避免直接编辑 sendmail.cf 文件。要对 sendmail 作配置修改，请编辑/etc/mail/sendmail.mc 文件，并备份原/etc/mail/sendmail.cf，可用以下方法生成新配置文件：

- 1) 使用/etc/mail/(make all -C /etc/mail/)包含的 makefile 来创建一个新的 /etc/mail/sendmail.cf 配置文件。

/etc/mail 里生成的其他文件，在必要时可以重新生成。仍然可以使用老的 makemap 命令。Service sendmail start | restart | reload 会自动使用这个命令。

- 2) 或者你可以使用 m4 处理器来创建一个新的/etc/mail/sendmail.cf。

m4 宏处理器不是默认安装的。使用它创建/etc/mail/sendmail.cf 前，请作为 root 用户安装 m4：

```
yum install m4
```

各种 Sendmail 配置文件都安装在/etc/mail/目录里，包括：

- 1) Access

指明可以使用 sendmail 传送邮件的系统。

- 2) Domaintable

指明主机名。

- 3) local-host-names

指明主机别名。

- 4) mailertable

指明对特定域重载路由的指令。

- 5) virtusertable

指定某领域特定形式的别名，该别名允许一个机器上可以驻留多个虚拟域。

在 Sendmail 能使用任何配置变更前，/etc/mail/里的很多配置文件(如 access、

domaintable、mailtable 和 virtusertable)必须实际在数据库文件中储存他们的信息。若要包含对他们数据库文件的这些配置所做的变更,请作为更用户运行下列命令:

```
makemap hash /etc/mail/<name> < /etc/mail/<name>
```

<name>代表被更新的配置文件名。通过运行下列命令,你也可以重启 sendmail 服务来使配置修改生效:

```
service sendmail restart
```

例如,若要所有 example.com 域的邮件地址都被发送到 bob@other-example.com,请在 virtusertable 文件里增加以下几行:

```
@example.com bob@other-example.com
```

为最终完成配置修改,必须更新 virtusertable.db 文件。

```
makemap hash /etc/mail/<name>
```

Sendmail 会创建被更新的 virtusertable.db 文件,该文件包含新的配置。

5.3.2.3 常见 Sendmail 配置变化

当更改 Sendmail 配置文件,最好不要编辑现有的文件,而是生成全新的 /etc/mail/sendmail.cf 文件。

 **警告:** 在改变配置内容前,请备份 sendmail.cf 文件:在改变 sendmail.cf 文件前,最好创建一个备份。

若希望给 Sendmail 增加需要的功能,请作为 root 用户编辑 /etc/mail/sendmail.mc 文件,一旦完成编辑,请重启 sendmail 服务:

```
service sendmail restart
```

如果安装了 m4 软件包,m4 宏处理器将自动生成新的 sendmail.cf 配置文件。
将 Sendmail 配置为其他客户端的服务器:

默认的 `sendmail.cf` 文件只允许 Sendmail 接受来自于本地机器的网络连接，而不允许 Sendmail 接受来自于任何其他主机的网络连接。若要将 `sendmail` 配置为其他客户端的服务器，请编辑 `/etc/mail/sendmail.mc` 文件，请改变 `DAEMON_OPTIONS` 指令的 `Addr=` 选项中指定的地址，从 `127.0.0.1` 地址改为主动网络设备的 IP 地址，或者通过在此行起始位置添加 `#` 来注释掉 `DAEMON_OPTIONS` 指令。结束后，通过重启服务重新生成 `/etc/mail/sendmail.cf`：

```
service sendmail restart
```

中标麒麟高级服务器操作系统的默认配置对多数只用 SMTP 的网站来说都是有效的然而，对于使用 UUCP (UNIX-to-UNIX Copy 协议) 的站点则不然。若使用 UUCP 邮件传输程序，则必须配置 `/etc/mail/sendmail.mc` 文件，并生成一个新的 `/etc/mail/sendmail.cf` 文件。

在编辑 `/usr/share/sendmail-cf` 目录下子目录中任何文件前，请查看 `/usr/share/sendmail-cf/README` 文件，因为那些文件能影响到 `/etc/mail/sendmail.cf` 文件将来的配置。

5.3.2.4 伪装

一个通用的 Sendmail 配置要具备一个机器，此机器作为网络上所有机器的邮件网关。例如，一个公司要有一个名为 `mail.example.com` 的机器，此机器处理他们的所有邮件，并为所有发出邮件指定一个一致的返回地址。

这种情况下，Sendmail 服务器必须伪装公司网络上的机器名，从而使他们的返回地址是 `user@example.com`，而不是 `user@host.example.com`。

要完成这一过程，请向 `/etc/mail/sendmail.mc` 添加以下行：

```
FEATURE(always_add_domain)dnl
FEATURE(`masquerade_entire_domain')dnl
FEATURE(`masquerade_envelope')dnl
FEATURE(`allmasquerade')dnl
MASQUERADE_AS(`bigcorp.com.')dnl
MASQUERADE_DOMAIN(`bigcorp.com.')dnl
MASQUERADE_AS(bigcorp.com)dnl
```

用 `m4` 宏处理器生成新 `sendmail.cf` 后，此配置使来自于网络内的所有邮件看

上去好像都是从 bigcorp.com 发出的。

5.3.2.5 阻止垃圾邮件

垃圾邮件可被定义为，未经用户请求时用户接收到的不必要的和有害的邮件。垃圾邮件造成了互联网标准的普遍滥用。

Sendmail 使抵制新的垃圾邮件发送技术相对容易。默认情况下，它抵制很多常用的垃圾邮件发送方法。Sendmail 中主要的反垃圾邮件功能 Main 是 are 标题检查，拒绝转播（从版本 8.9 就默认此功能），访问数据库和信息检查。

例如，自 Sendmail 版本 8.9 起，已默认禁用转发 SMTP 信息(也称为转播)。在发生这种变化之前，Sendmail 指示邮件主机(x.edu)接收来自于一方(y.com)的邮件，并将他们发送到不同方(z.net)。然而，现在 Sendmail 必须被配置为允许任何域通过服务器来转播邮件。要配置转播域，请编辑 A/etc/mail/relay-domains 文件，并重启 Sendmail。

```
service sendmail restart
```

然而，很多时候，用户遭受来自于互联网上其他服务器的垃圾邮件轰炸。这种情况下，/etc/mail/access 文件中可用的 Sendmail 的访问控制功能可用于防止来自于有害主机的连接。下例描述了如何使用这个文件来阻止访问 Sendmail 服务器，以及如何使用这个文件来允许访问 Sendmail 服务器。

```
badspammer.com ERROR:550 "Go away and do not spam us anymore"
tux.badspammer.com OK 10.0
RELAY
```

本例表明，从 badspammer.com 发出的任何邮件都被阻止，给出了符合 550 RFC-821 错误代码，并向滥发垃圾邮件者回送一个信息。从 tux.badspammer.com 子域发出的邮件被接受。最后一行表明从 10.0.*.*网络发出的任何邮件都可被邮件服务器转播。

因为/etc/mail/access.db 文件是数据库，所以请使用 makemap 命令来更新任何改变。要这样做，请作为 root 用户使用以下命令：

```
makemap hash /etc/mail/access < /etc/mail/access
```

信息标题分析使您可以基于标题的内容来拒绝邮件。SMTP 服务器在信息标

题中存储一个邮件的行程信息。当信息从一个 MTA 到达另一个 MTA, 每个 MTA 在所有其他 Received 标题之上提交一个 Received 标题。请注意, 这个信息可能会被垃圾邮件发送者改变。

上面的示例仅代表 Sendmail 中关于允许或阻止访问功能的一小部分。更多信息和示例, 请参考/usr/share/sendmail-cf/README。

因为投递邮件时 Sendmail 调用 the Procmail MDA, 也可以使用垃圾邮件过滤程序 (如 SpamAssassin) 来为用户识别并归档垃圾邮件。更多关于使用 SpamAssassin 的信息, 请参考 5.4.2.6 垃圾邮件过滤器。

5.3.2.6 使用 Sendmail 与 LDAP

要从大量用户中查找特别用户的特定信息, 使用 LDAP 是非常快速和强大的方法。例如, LDAP 服务器可以用来通过用户名字从普通的目录里查找个别邮件地址。在实施过程中, LDAP 很大程度上独立于 Sendmail, LDAP 储存分层用户信息, 而只有 Sendmail 能得到自动填写收信人的邮件信息中的 LDAP 查询结果。

然而, Sendmail 支持与 LDAP 更加紧密的集成方式, 在这种集成中, Sendmail 使用 LDAP 来代替不同邮件服务器上独立保存的文件 (如/etc/aliases 和 /etc/mail/virtusertables), 这些服务器协同工作, 为一个中等-企业级的组织提供支持。简而言之, LDAP 从 Sendmail 中提取邮件路由层次, 这些路由层次可能来自于其单独的配置文件, 也可能来自于被许多不同的应用程序所影响的强大 LDAP 集群。

sendmail 的目前版本包括对 LDAP 的支持。若要使用 LDAP 扩展 sendmail 服务器, 请首先获取正在运行并且配置正确的 LDAP 服务器, 例如 OpenLDAP。然后编辑/etc/mail/sendmail.mc, 使其包括以下信息:

```
LDAPROUTE_DOMAIN('yourdomain.com')dnl
LDAPROUTE_DOMAIN('yourdomain.com')dnl
```



提示: 高级配置:

这只是针对有 LDAP 的 sendmail 的基本配置。根据 LDAP 的实施方法的不同, 配置也有很大区别, 特别是当配置几个 Sendmail 机器使用公共 LDAP 服务器时。

详细的 LDAP 路由配置说明和示例, 请参考/usr/share/sendmail-cf/README。

下一步，通过运行 `m4` 宏处理器以及重启 `sendmail` 来重新创建 `/etc/mail/sendmail.cf`。具体说明，参考 5.3.2.3 常见 `sendmail` 配置改变。

5.3.3 邮件工具（Fetchmail）

邮件工具是一个 MTA，MTA 从远程服务器检索邮件并将邮件投递到本地 MTA。很多用户倾向于把下载远程服务器上信息的过程与在 MUA 中阅读和组织邮件的过程分开。根据用户拨号需求而设计，通过使用任何数量的协议（包括 POP3 和 IMAP），邮件工具连接并将所有邮件信息快速下载到邮件假脱机文件。必要时，它能将邮件信息转发到 SMTP 服务器上。

安装 Fetchmail 包：

要使用邮件工具，首先要确保您的系统已安装 Fetchmail 包，若要安装，请作为更用户运行如下命令：

```
yum install fetchmail
```

通过使用用户主目录中的 `.fetchmailrc` 文件，可以为每个用户配置邮件工具。如果该文件不存在，请在你的主目录中创建 `.fetchmailrc` 文件。

使用 `.fetchmailrc` 文件的参数选择，邮件工具查找并下载远程服务器上的邮件。然后将邮件投递到本地机器的端口 25，使用本地 MTA 将邮件放在正确的用户邮件假脱机文件里。如果 Procmail 可用，将启动 Procmail 过滤邮件并放入收件箱，那样 MUA 就可以阅读邮件。

5.3.3.1 邮件工具配置选项

虽然运行邮件工具时，可以通过在命令行上传递所有必要的选项来检查远程服务器上的电子邮件，但是使用 `.fetchmailrc` 文件更容易。将您期望的配置选项放在 `.fetchmailrc` 文件中，从而每次发出 `fetchmail` 命令时都使用这些选项。通过在命令行中指定那个选项来运行邮件工具时，也可以重载这些选项。

用户的 `.fetchmailrc` 文件包含 3 类配置选项：

1) global options

给出邮件工具指令，这些邮件工具指令或者控制程序运行，或者为每个查看邮件的连接提供设置。

2) server options

描述指定服务器的必要信息（如主机名）以及特定邮件服务器的参数（如检查的端口号，或超时前所需等待的秒数）。这些选项影响每个使用此服务器的用户。

3) user options

包含使用特定邮件服务器时需要认证和查找邮件的必要信息（如用户名和密码）。

全局选项出现在`.fetchmailrc` 文件顶端，后面跟着一个或多个服务器选项，其中每个选项指定邮件工具应该查看的不同电子邮件服务器。用户选项排列在服务器选项的后面，用于为每个用户账户对那个电子邮件服务器进行查看。像服务器选项一样，可以指定多个用户选项来使用特定服务器，以及检查同一服务器上的多个邮件账户。

通过使用特殊的选项动词 `poll` 或 `skip`，服务器选项被`.fetchmailrc` 文件中的服务所调用，选项动词在任何服务器信息之前。`poll` 动作通知邮件工具在运行时使用此服务器选项，这个动作使用特定的用户选项来查找电子邮件。`Skip` 动作后的任何服务器选项都不会被检查，除非在邮件工具被调用时指定了这个服务器的主机名。在测试`.fetchmailrc` 文件里的配置时，`skip` 选项非常实用，因为当被明确的调用时，它只检查被跳过的服务器，不影响当前的工作配置。

以下是`.fetchmailrc` 文件的一个示例：

```
set postmaster "user1"
set bouncemail

poll pop.domain.com proto pop3
user 'user1' there with password 'secret' is user1 here

poll mail.domain2.com
user 'user5' there with password 'secret2' is user1 here
user 'user7' there with password 'secret3' is user1 here
```

在这个例子里，全局选项说明用户正在尝试最后一次(`postmaster` 选项)发送电子邮件，所有电子邮件错误被发送给邮局主管而不是发送者(`bouncemail` 选项)。`set` 动作通知邮件工具此行包含一个全局选项。然后，就指定了两个邮件服务器，其中一个设置为使用 `POP3` 检查的工作方式，另一个被设置为尝试从各种协议中

找到可用协议的工作方式。使用第二个服务器选项来检查这两个用户，但是为任何用户检查到的所有邮件都将被发送到 `user1` 的邮件池里。这使得我们可以在多个服务器上检查多个邮箱，但其只出现在单个 MUA 收件箱中。每个用户特定信息都从 `user action` 开始。



提示：省略配置中的密码：

用户不允许把密码放到 `.fetchmailrc` 文件里。省略 `with password '<password>'` 部分导致邮件工具开启时需要密码。

邮件工具有很多全局选项、服务器选项和本地选项。其中有很多选项都很少用到，或者只适用于非常特定的情况。`Fetchmail` 页面对每个选项做详细解释，下面 3 节中将为您列出最通用的选项。

5.3.3.2 全局选项

每个全局选项都必须放在每行的 `Set` 动作后。

1) `daemon <seconds>`

指定守护进程模式，该模式下邮件工具在后台运行。用查询服务器邮件工具要等待的秒数来代替 `<seconds>`。

2) `postmaster`

如果投递有问题，指定本地用户发送邮件。

3) `syslog`

指定错误或状态信息的日志文件。默认情况下，这个文件是 `/var/log/maillog`。

5.3.3.3 服务器选项

服务器选项必须被放在 `.fetchmailrc` 中 `poll` 或 `skip` 动作后的相应行内。

4) `auth<auth-type>`

以所使用的认证类型代替 `<auth-type>`。默认情况下，使用密码认证，但是一些协议支持其他类型的认证，包括 `kerberos_v5`，`kerberos_v4`，和 `ssh`。如果使用了 `any` 认证类型，邮件工具首先尝试不需要密码的方法，然后加密密码，并最终发送加密密码到认证服务器。

5) `interval<number>`

每检查 `<number>` 次所有已配置服务器上的电子邮件，就轮询一次指定的服务器。这个选项通常用于用户很少接收邮件的邮件服务器。

6) port<port-number>

用端口号代替<port-number>。默认情况下，为指定协议重载这个数值。

7) proto<protocol>

用协议代替<protocol>，如 POP3 或者用来在服务器上检查信息的 imap。

8) timeout<seconds>

用服务器静止的秒数来代替<seconds>，服务器静止后，邮件工具放弃连接尝试。如果没有设置这个数值，默认值为 300 秒。

5.3.3.4 用户选项

用户选项可以放在服务器选项下它本身所在的那一行，或者与服务器选项在同一行。每种情况下，定义的选项必须遵循用户选项（如以下定义）。

1) Fetchall

邮件工具下载队列中信息的顺序，包括已经浏览过的信息。默认情况下，邮件工具只能下拉显示新信息。

2) fetchlimit<number>

以停止之前检索到的信息数量代替<number>。

3) flush

在检索新信息之前，删除以前浏览过的队列中的信息。

4) limit<max-number-bytes>

以用邮件工具检索时所允许信息的最大字节数代替<max-number-bytes>。当需要很长时间而不能下载大邮件时，这个选项对慢速网络连接有效。

5) password<password>

以用户密码代替<password>。

6) preconnect<command>

以为用户检索邮件后执行的命令代替<command>。

7) ssl

激活 SSL 加密。

8) user<username>

以邮件工具检索邮件所使用的用户名代替<username>。这个选项必须优先于所有其他用户选项。

5.3.3.5 邮件工具命令选项

当执行邮件工具命令时，大多数用于命令行的邮件工具选项都反映`.fetchmailrc` 的配置选项。这种情况下，邮件工具可需要或不需配置文件。多数用户不在命令行中使用这些选项，因为把这些选项放在`.fetchmailrc` 文件中更容易。

有时候为了特殊目的，希望用其他选项来运行 `fetchmail` 命令。也可以发布命令来临时重载发生错误的`.fetchmailrc` 设置，因为命令行中指定的任何选项都重载配置文件选项。

5.3.3.6 信息或调试选项

`Fetchmail` 命令后的某些选项可以提供重要信息。

1) `-configdump`

显示基于`.fetchmailrc` 信息和邮件工具默认信息的每个可能选项。使用这个选项时，不为任何用户检索邮件。

2) `-s`

执行邮件工具的安静模式，防止任何信息，或其他错误在 `fetchmail` 命令后发生。

3) `-v`

执行邮件工具的详细模式，显示邮件工具与远程邮件服务器之间的每次通讯。

4) `-V`

显示详细的版本信息，列举全局选项，并且显示每个用户使用的设置，包括邮件协议和认证方法。使用这个选项时，不为任何用户检索邮件。

5.3.3.7 特殊选项

这些选项偶尔对重载常常能在`.fetchmailrc` 文件里发现的默认信息非常有用。

1) `-a`

邮件工具从远程邮件服务器下载所有信息，无论是新的还是浏览过的。默认情况下，邮件工具只能下拉新信息。

2) `-k`

下载后，邮件工具把信息留在远程邮件服务器上。这个选项重载下载信息后删除信息的默认行为。

3) `-l<max-number-bytes>`

邮件工具不下载超过特定大小的信息，并把这种信息留在远程服务器上。

4) -quit

取消邮件工具守护进程程序。

Fetchmail 页面可以看到更多命令和fetchmailrc 选项。

5.3.4 邮件传输代理（MTA）配置

邮件传输代理（MTA）对发送邮件来说是非常必要的。邮件传输代理如 Evolution、Thunderbird 和 Mutt 是用来读取和编辑邮件的。当用户从 MUA 发送邮件时，信息被转交给 MTA，MTA 通过一系列的 MTA 来发送信息，直到信息到达目的地。

即使用户不打算从系统发送邮件，有些自动的任务或系统程序也会用 /bin/mail 命令向本地系统的 root 用户发送包括登陆信息的邮件。

中标麒麟高级服务器操作系统提供两种 MTA—Postfix 和 Sendmail，如果都安装，Postfix 为默认 MTA。

5.4 邮件投递代理

中标麒麟高级服务器操作系统包括两种主要 MDA，Procmail 和 mail。两种应用都视作 LDA，并且两种都从 MTA 文件移动邮件到用户的邮箱。然后，Procmail 可以提供强大的过滤系统。

此部分仅描述 Procmail 信息，Mail 命令的更多信息，参考主页(man mail)。

当邮件位于本地主机的邮件假脱机文件中时，Procmail 投递和过滤电子邮件。它很强大，只占有很少的系统资源，被广泛应用。在投递邮件过程中 Procmail 起到非常重要的作用，它使邮件客户端应用程序可以读取邮件。

可以通过不同方法调用 Procmail。无论何时把 MTA 放入邮件假脱机文件，procmail 都将被启动。然后 Procmail 为 MUA 过滤和归档邮件，并退出。或者，可以将 MUA 配置为：任何时候只要收到信息就执行 Procmail，从而信息都被移入正确的邮箱。默认情况下，无论 MTA 何时收到新信息，出现在用户主目录下的/etc/procmailrc 或 ~/.procmailrc 文件都调用 Procmail。

默认情况下，没有系统级的文件存在于/etc/目录，也没有 no.procmailrc 文件存在于用户的主目录。因此，要使用 Procmail，每个用户必须用特定环境变量和规则构造一个.procmailrc 文件。

Procmail 是否对电子邮件信息起作用取决于信息是否与 rc 文件中特定的条件集或规则集相匹配。如果一个信息与一个规则相匹配，则电子邮件或被放置在指定文件中，或被删除，或做其他处理。

当 Procmail 启动时，它读取电子邮件信息，并把邮件体和标题信息分开。然后，Procmail 寻找/etc/procmailrcs 目录中的/etc/procmailrcs 文件和 rc 文件，从中获取默认的系统级 Procmail 环境变量和规则。然后 Promail 在用户主目录中搜寻.procmailrc 文件。很多用户参考他们主目录中的.procmailrc 文件，从而为 Promail 额外创建 rc 文件。

5.4.1 Procmail 配置

Procmail 配置文件包括重要的环境变量。这些变量指明哪些信息需要排序，以及如何处理不匹配任何规则的信息。

这些环境变量通常以下列模式出现在 ~/.procmailrc 文件的开始：

```
<env-variable>=<value>
```

本例中<env-variable>是变量名，<value>定义了变量。

大部分 procmail 的用户使用许多环境变量。重要的环境变量在默认值中已经设置。大部分情况下，使用以下变量。

1) DEFAULT

设置默认收件箱，将所有与任何规则都不匹配的信息都放在此邮箱中。默认的错误值和 SORGMAIL 相同。

2) INCLUDERC

指定额外 rc 文件，它包含更多检查信息的规则。这将 Procmail 规则列表划分为单个文件，这些文件履行不同的角色，如阻塞垃圾邮件，处理邮件列表，然后在用户的 ~/.procmailrc 文件中可以通过使用注释符号来将此功能设置为关或开。

例如，一个用户的 ~/.procmailrc 文件可能看上去如下所示：

```
MAILDIR=$HOME/Msgs
INCLUDERC=$MAILDIR/lists.rc
INCLUDERC=$MAILDIR/spam.rc
```

要关闭 procmail 邮件过滤但是保留垃圾邮件控制功能，请用#注释掉第一行，

即 INCLUDERC 行。

3) LOCKSLEEP

以秒为单位设定时间段：Procmail 企图使用特定 lockfile 的时间间隔，默认为 8 秒。

4) LOCKTIMEOUT

以秒为单位设定时间段：自最后一次修改 lockfile 后，Procmail 认为 lockfile 过期且可以被删除之前必须经历的时间段。默认为 1024 秒。

5) LOGFILE

Procmail 信息或错误消息被写入的文件。

6) MAILDIR

为 Procmail 设置当前工作目录。若已设置，所有其他 Procmail 路径将与此目录相关联。

7) ORGMAIL

如果邮件不能被放在默认的或规则所需的位置，使用此变量指定源邮箱或另一个存放信息的地方。默认情况下，使用/var/spool/mail/\$LOGNAME 的一个值。

8) SUSPEND

以秒为单位，设置不能获得必要资源（如交换空间）时，Procmail 暂停的时间。

9) SWITCHRC

允许用户指定包含其他 Procmail 规则的外部文件，该选项与 INCLUDERC 选项基本相同，他们的区别在于，在访问规则配置文件时就停止规则检查，并且只用到 SWITCHRC 特定文件中的规则。

10) VERBOSE

使 Procmail 将更多信息记入日志，这个选项对调试很有用。

shell 上列出了其他重要的环境变量，如 LOGNAME，登录名称；Home 主目录位置，以及默认的 SHELL。

5.4.2 Procmail Recipes

新用户经常会发现，构建邮件检查规则是使用 Procmail 中最难于学习的部分。某种程度上，这是可以理解的，因为这些规则是使用正则表达式来完成信息

匹配的，正则表达式是用于描述字符串匹配条件的特殊格式。即使这样，正则表达式并不是非常难于构建，甚至阅读正则表达式时也不那么困难。另外，一致的 Procmail 规则编写方式，使得通过范例学习 Procmail 规则更为容易，而不考虑正则表达式的要求。要查看 Procmail 的例子，请参考 5.4.2.5 规则示例。

Procmail 规则采用如下形式：

```
:0<flags>: <lockfile-name> * <special-condition-character>
<condition-1> * <special-condition-character>
<condition-2> * <special-condition-character>
<condition-N>
<special-action-character>
<action-to-perform>
```

Procmail 规则的前两个字符是一个冒号和一个零。各种标记可被放在 0 后面来控制 Procmail 处理规则的方式。<flags>后面的冒号说明为这个信息创建了一个锁定文件。创建了一个锁定文件，通过替换<lockfile-name>可以指定文件名。

一个规则可以包含几个与信息进行匹配的条件。如果它没有任何条件，则每一条信息都与规则相匹配。将正则表达式放置在某些条件中，这可以促进信息匹配。如果使用了多个条件，他们必须与要执行的动作全部匹配。基于规则第一行的标记集合对条件进行检查。星号符号(*)后的可选特殊字符能进一步控制条件。

<action-to-perform>参数说明当信息与一个条件相匹配时要执行的动作。每个规则可能只有一个动作。多数情况下，邮箱名在这里用于将匹配信息写入那个文件，并有效的排序邮件。指定动作前，可能会采用特殊的动作。更多信息，请参考 5.4.2.4 特殊条件与动作。

5.4.2.1 投递和非投递规则

判断规则与特定信息是否相匹配时所使用的动作，决定了此规则是投递规则还是非投递规则。投递规则包含一个动作，该动作向文件写入信息，或向另一程序发送信息，或将信息转发到另一个邮件地址。非投递规则包括其他动作，如嵌套程序。嵌套程序是包含在括号{ }中的一系列动作，系统在与规则条件相匹配的信息上执行嵌套程序。嵌套程序可以相互嵌套，这为识别并执行消息上的动作提供了更强的控制措施。

当信息与投递规则匹配时，Procmail 执行特定的动作，并停止信息与其他规

则的比较。而非投递规则相匹配的信息继续与其他规则进行比较。

5.4.2.2 标记

若要决定规则的条件如何与信息相比较,或是否与信息相比较,则有必要使用标记。下面是一些常用标记。

1) A

说明: 仅当前一个没有 A 或 a 标记的规则也与该信息相匹配时,才使用此规则。

2) a

说明: 仅当前一个具有 A 或 a 标记的规则也与该信息相匹配,并且完成成功匹配时,才使用此规则。

3) B

解析信息体并查找能匹配的条件。

4) b

在任何结果动作中使用此标记,如向文件中写入此信息或转发此信息的动作,这是默认的。

5) c

生成电子邮件副本。这对投递规则来说是非常有用的,因为能对信息执行所需要的动作,并且可以继续在此 rc 文件中处理信息的副本。

6) D

使 egrep 比较区分大小写。默认情况下, egrep 比较过程并不区分大小写。

7) E

与 A 标记相似,仅当前一个没有 E 标记的规则不匹配时,规则中的条件才与信息相比较。这犹如一个 else 动作。

8) e

仅当前一个规则匹配失败时,才比较此规则与信息。

9) f

使用管道作为过滤器。

10) H

解析信息标题,并查找相匹配的条件。这是默认的。

11) h

使用结果动作中的标题。这是默认的。

12) w

通知 Procmail 等待指定过滤器或程序完成运行，并在关注被过滤的信息前报告是否运行成功。

13) W

与 w 基本相同，区别之处在于"Program failure"信息被禁止。

5.4.2.3 识别本地锁定文件

锁定文件对 Procmail 是非常实用的，它可以确保多个（多于一个）过程不能同时对信息进行改变。通过在规则首行的任何标记后放置一个冒号(:)，从而指定一个本地锁定文件。这种方法创建了一个本地锁定文件，此文件基于目标文件名字和 LOCKEXT 全局环境变量中设置的内容。

另外，也可以在冒号后指定此规则的本地锁定文件名。

5.4.2.4 特殊条件与动作

Procmail 规则条件和动作前的特殊字符，将改变对 Procmail 规则条件和动作的解释。

在一个规则条件行的开始处，以下字符可用于星号符号后(*)：

1) !

在条件行中，这个符号将反转条件，在发生条件与信息不匹配时，反转后的条件将引发一次匹配。

2) <

检查信息容量是否少于指定的字节数。

3) >

检查信息容量是否多于指定的字节数。

下面的符号可用于执行特殊的动作。

4) !

在动作行中，此符号通知 Procmail 将信息转发到指定邮箱地址。

5) \$

引用早在 rc 文件中设置的变量。这个符号常用于设置被各种规则引用公共邮箱。

6) |

启动一个特定程序来处理信息。

7) {and}

构造嵌套程序，用于包含更多应用于匹配信息的规则。

若在动作行的起始处没有任何特殊符号，则 **Procmail** 假定那个动作行指定了编写邮件所用的邮箱。

5.4.2.5 规则示例

Procmail 是一个极其灵活的程序，但正是由于这种灵活性，使得对于新手来说，构造 **Procmail** 规则是一件很难的事情。

学习构造 **Procmail** 规则条件技术的最佳方式就是，深入理解正则表达式，并分析很多其他开发人员构造的 **Procmail** 规则条件示例。关于正则表达式的详细解释超出了本节的范围。在互联网上的很多地方（如 <http://www.iki.fi/era/procmail/links.html>），您都能找到 **Procmai** 规则的结构以及有用的 **Procmail** 规则示例。通过查看这些规则示例，促使您合适的使用和改编正则表达式。另外，在 **grep** 手册页您可以发现关于基本正则表达式规则的介绍性信息。

下面的简单示例描述了 **Procmail** 规则的基本结构，这些示例为更复杂的构造提供了基础。

一个基本的规则可能不包含条件，如下例所示：

```
:0: new-mail.spool
```

第一行指定了要被创建的本地锁定文件，但没有指定文件名，所以 **Procmail** 使用目标文件名，并附加在 **LOCKEXT** 环境变量中指定的值。因为没指定任何条件，所以每个信息与规则都是匹配的，并被放入一个叫做 **new-mail.spool** 的文件池，该文件池位于 **MAILDIR** 环境变量所指定的目录中。然后 **MUA** 可以查看此文件中的信息。

一个如上所示的基本规则，可以放在所有 **.rc** 文件的尾部，用来将信息放入到默认的位置。

下例与指定邮件地址的信息相匹配，并被丢弃。

```
:0 * ^From: spammer@domain.com /dev/null
```

本例中，由 `spammer@domain.com` 发送的任何信息都将被发送至 `/dev/null` 设备，并删除它们。



警告：发送信息至 `/dev/null`

在向 `/dev/null` 发送永久删除信息指令前，请确认规则正在以规定的方式运行。若一个规则无意间获取了非计划中的信息，而那些信息又消失了，要修复此规则将变得非常困难。

更好的解决方法是将规则的动作指向一个指定的邮箱，可不时检查该邮箱来查找错误信息。一旦发现无信息偶然匹配，执行将信息发送至 `/dev/null` 的动作。

以下规则截取由特定邮件列表发出的邮件，并将其放入指定文件夹：

```
:0: * ^((From|Cc|To).*)tux-lug tuxlug
```

由 `tux-lug@domain.com` 邮件列表发出的任何信息都将为 MUA 自动放在 `tuxlug` 邮箱中。注意，若在 `From`、`Cc` 或 `To` 行中有邮件列表中的邮箱地址，则本例中的条件与信息是相匹配的。

更多详细及强大的规则，参见 5.6 更多信息资源中的很多 Procmail 在线资源。

5.4.2.6 垃圾邮件过滤器

当收到新邮件时，Sendmail, Postfix 及 Fetchmail 均调用垃圾邮件过滤器，所以 Procmail 可用作拦截垃圾邮件的强大工具。

当 Procmail 与 SpamAssassin 结合使用时，其功能尤其强大。当这两个程序同时使用时，他们能很快的识别垃圾邮件，过滤并毁坏他们。

SpamAssassin 通过标题分析、文本分析、黑名单、垃圾邮件搜寻数据库及自学习 Bayesian 垃圾邮件分析快速准确的识别并标记垃圾邮件。



提示：安装 spamassassin 程序包：

要使用 SpamAssassin，首先应确保已将 spamassassin 程序包安装至您的系统，可以通过作为 root 用户运行以下命令来完成安装：

```
yum install spamassassin
```

本地用户使用 SpamAssassin 的更简便的方法是将以下行放至 ~/.procmailrc 文件开头附近。

```
INCLUDERC=/etc/mail/spamassassin/spamassassin-default.rc
```

/etc/mail/spamassassin/spamassassin-default.rc 文件包含一个激活 SpamAssassin 过滤所有接收邮件的简单的 procmail 规则。若一封邮件被判定为垃圾邮件，如下标记它的标题：

```
*****SPAM*****
```

邮件的信息体也将带有一个表明何种因素导致该邮件被判定为垃圾邮件的运行标签。

要对标示为垃圾邮件的邮件归档，可使用类似于如下格式的规则：

```
:0 Hw * ^X-Spam-Status: Yes spam
```

一旦一个邮件的标题被标记为垃圾邮件，则该规则就将其归入名为 spam 的邮箱。

因 SpamAssassin 为 Perl 脚本，也许需在已被占用的服务器上使用二进制 SpamAssassin 守护程序 (spamd) 及客户应用程序(spamc)。然而，按此方法设置 SpamAssassin 需用户具备 root 用户级别访问权限。

要运行 spamd 守护程序，请输入如下指令

```
service spamassassin start
```

要在系统启动时就运行 SpamAssassin 守护程序，请使用初始化脚本，如 Services Configuration Tool (system-config-services)来启用 spamassassin 服务。

要设置 Procmail 使用 SpamAssassin 客户端工具（而非 Perl 脚本），请将以下行放至 ~/.procmailrc 文件开头附近。对全系统的设置，将其放至/etc/procmailrc 中。

```
INCLUDERC=/etc/mail/spamassassin/spamassassin-spamc.rc
```

5.5 邮件用户代理

中标麒麟高级服务器操作系统提供多种邮件程序，既有图形化电子邮件客户端程序（如 Thunderbird），也有基于文本的邮件程序（如 mutt）。

本章节其余部分着重讲述客户端及服务器间的安全通信。

5.5.1 安全通信

中标麒麟高级服务器操作系统中配备受欢迎的 MUAs，如 Thunderbird 与 mutt 提供 SSL 加密电子邮件会话。

如其他广泛用于未加密网络的服务一样，重要邮件信息（如用户名、密码及整个信息）可被网络上的其他用户拦截并浏览。此外，因标准的 POP 及 IMAP 协议传递未加密的认证信息，当用户名和密码在网络上被传递时，黑客有可能通过收集这些用户名及密码（因这些信息在整个网络中备传递）从而获取用户账号。

5.5.1.1 安全邮件客户端

多数用于查找远程服务器上的邮件 LinuxMUA 都支持 SSL 加密。若要在检索邮件时使用 SSL，必须同时在邮件客户端及服务器端启动 SSL。

在客户端上启用 SSL 是很容易的，通常通过单击 MUA 配置窗口中的一个按钮，或 MUA 配置文件中的一个选项即可完成。安全 IMAP 与 POP 带有 MUA 用于认证及下载信息的已知端口号（分别为 993 和 995）。

5.5.1.2 安全邮件客户端通信

向邮件服务器上的 IMAP 和 POP 用户提供 SSL 加密是件简单的事。

首先，创建一个 SSL 证书。可通过两种方法完成：向 Certificate Authority (CA) 申请一个 SSL 证书或创建一个自签名证书。

 **警告：**避免使用自签名证书：

自签名证书仅可用于检测目的。生产情况下的任何服务器均应使用 CA 给予的 SSL 证书。

若要为 IMAP 或 POP 创建一个自签名 SSL 证书，请转换至 /etc/pki/dovecot/ 目录，并根据您的需要编辑 /etc/pki/dovecot/dovecot-openssl.conf 配置文件中的证书参数，然后作为 root 用户输入如下命令：

```
rm -f certs/dovecot.pem private/dovecot.pem
/usr/libexec/dovecot/mkcert.sh
```

完成后，请确保您已在/etc/dovecot/conf.d/10-ssl.conf 文件中做了如下配置：

```
ssl_cert = </etc/pki/dovecot/certs/dovecot.pem
ssl_key = </etc/pki/dovecot/private/dovecot.pem
```

执行 `service dovecot restart` 指令来重启 dovecot 守护程序。

或者，也可将 `stunnel` 指令可以作为 IMAP 或 POP 服务器标准不安全连接上的 SSL 加密包装器。

`Stunnel` 实用工具利用中标麒麟高级服务器操作系统自带的外部 `OpenSSL` 库，提供强大的密码使用法并保护网络连接。推荐向 CA 申请来获取 SSL 证书，但也可创建一个自签名证书。



提示：安装 `stunnel` 软件包：

若要使用 `stunnel`，首先作为 `root` 用户运行以下命令，确保已将 `stunne` 软件包安装至您的系统。

```
yum install stunnel
```

要创建一个自签名证书，请转换至/etc/pki/tls/certs/目录，输入如下命令：

```
make stunnel.pem
```

回答所有问题从而完成整个程序。

一旦生成了证书，创建 `stunnel` 配置文件，例如包含以下内容的/etc/stunnel/mail.conf：

```
cert = /etc/pki/tls/certs/stunnel.pem

[pop3s]
accept  = 995
connect = 110

[imaps]
accept  = 993
connect = 143
```

使用 `/usr/bin/stunnel/etc/stunnel/mail.conf` 命令，用已创建配置文件启动

stunnel，一旦上述操作完成，您即可使用 IMAP 或 POP 邮件客户端，并将此客户端使用 SSL 加密连接至邮件服务器。

有关 stunnel 的更多信息，参见 stunnel 帮助页面或/usr/share/doc/stunnel-<version-number>/目录中的文件，其中，<version-number>为 stunnel 的版本号。

5.6 更多信息资源

以下是关于邮件应用程序的更多文件资料。

5.6.1 已安装文件

1) 包含设置 sendmail 的信息，以及 sendmail 与 sendmail-cf 软件包。

2) /usr/share/sendmail-cf/README

包含 m4 宏处理器、sendmail 文件地址、所支持邮件程序、如何获取高级功能及更多相关信息。

此外，sendmail 及 aliases 帮助页面包含很多有用信息，涵盖了多种 sendmail 选项信息及/etc/mail/aliases 文件正确配置的有用信息。

3) /usr/share/doc/postfix-<version-number>

包含大量 Postfix 配置方法的相关信息，用 Postfix 的版本号取代<version-number>。

4) /usr/share/doc/fetchmail-<version-number>

包含 FEATURES 文件中 Fetchmail 特色的完整列表及介绍性的 FAQ 文件。用 Fetchmail 的版本号取代<version-number>。

5) /usr/share/doc/procmail-<version-number>

包含关于 Procmail 综述的 README 文件，阐述每个程序功能的 FEATURES 文件，及解答很多常见配置问题的 FAQ 文件。用 Procmail 的版本号取代<version-number>。

在学习 procmail 工作方式及创建新规则时，下面 Procmail 帮助页面的内容是非常宝贵的。

1) Procmail

概述 Procmail 工作方式及过滤邮件的相关步骤。

2) Procmailrc

解释用于构件规则的 rc 文件格式。

3) Procmailex

提供若干 Procmail 规则的有用实例。

4) Procmailsc

解释加权打分技术，procmail 使用此技术将规则与信息进行匹配。

5) /usr/share/doc/spamassassin-<version-number>/

含有大量与 SpamAssassin 相关的信息。用 spamassassin 软件包的版本号取代<version-number>。

5.6.2 有用网址

<http://www.sendmail.org/> — 提供 Sendmail 特点、文档及配置示例的详细技术分析。

<http://www.sendmail.com/> — 提供关于 sendmail 的新闻、访谈及文章，包括多种选项的展开描述。

<http://www.postfix.org/> — Postfix 项目的首页，包含关于 postfix 的丰富信息。邮件列表是搜索信息的好地方。

<http://fetchmail.berlios.de/> — Fetchmail 的首页，以在线指导及详细的 FAQ 为特色。

<http://www.procmail.org/> — Procmail 首页，提供 procmail 归类邮件列表的链接及各种 FAQ 文件。

<http://partmaps.org/era/procmail/mini-faq.html> — 优秀的 procmailFAQ，提供故障解决技巧，文件锁的详细信息，以及通配符使用方法。

<http://www.uwasa.fi/~ts/info/proctips.html> — 涵盖使用 procmial 的许多技巧，这些技巧使 procmial 更容易使用。包括测试.procmailrc 文件的方式，以及如何使用 Procmail 打分来决定是否需采取特定措施。

<http://www.spamassassin.org/> — SpamAssassin 项目的官网。

5.6.3 相关书籍

Sendmail Milters: A Guide for Fighting Spam by Bryan Costales and Marcia Flynt; Addison-Wesley 一本优秀的 snedmail 指南，可帮助您自定义邮箱过滤器。

Sendmail by Bryan Costales with Eric Allman et al.; O'Reilly & Associates 一本好的 sendmail 参考书籍，是由 Delivermail 和 Sendmail 的创始人协助编纂的。

Removing the Spam: Email Processing and Filtering by Geoff Mulligan;
Addison-Wesley Publishing Company 本书着眼于邮件邮件管理员使用诸如
sendmail、Procmail 等工具来管理垃圾邮件的各种方法。

Internet Email Protocols: A Developer's Guide by Kevin Johnson ;
Addison-Wesley Publishing Company 本书全面概述了主要邮件协议及其所提供
的安全性。

Managing IMAP by Dianna Mullet and Kevin Mullet; O'Reilly & Associates 详
述配置 IMAP 服务器所需的步骤。

6 VSFTPD 配置和使用

6.1 关于 vsftpd

vsftpd 即 Secure FTP Daemon，是 Linux 系统中首选的快速、安全的 FTP 程序。它快速、稳定而具有高安全性，能够高效且安全地同时处理大量连接。

Vsftpd 使用的安全模式有以下三方面的特性：

- 1) privileged 和 non-privileged 程序的强大独立性；

不同的任务由各自独立的程序处理，而且它们在最小优先级的情况下运行。

- 2) 需要更高优先级的任务是由最小优先级的进程处理；

通过对兼容性的平衡，那些需要完全 root 优先权的任务可以由较低优先级的程序更安全的处理。

- 3) 绝大多数程序运行于 chroot jail。

一旦条件允许，程序被改变路径至共享目录，这个共享目录即被称为一个 chroot jail。

这些安全惯例在 vsftpd 处理请求时有以下一些影响，在很大程度上提高了系统的安全性能：

- 1) 父进程运行在所需要的最小优先级；

父进程可动态计算自己所需的优先级水平以降低风险；子进程同 FTP 客户端直接相互作用，它们尽可能运行在接近 0 优先级的水平。

- 2) 所有需要高优先级的操作会由一个小的父进程处理；

与 Apache HTTP Server 类似，vsftpd 启动一些没有优先级的子程序来处理进来的连接；这样可使父进程具有尽量低的优先级以处理相对较少的任务。

- 3) 父进程不信任由零优先级的子程序提交的请求；

父进程同子进程间通过套接字连接，子进程发来的任何信息的有效性在执行前都要通过检查。

- 4) 大多数同 FTP 客户端的交互是由在 chroot jail 中的没有优先级的子程序来处理。

因为没有优先级的子程序仅可以访问共享目录，即便被攻击，受损害的也只是共享目录。

6.2 Vsftpd 的常用配置文件

vsftpd RPM 包用于安装程序(/usr/sbin/vsftpd)及其相关配置文件，同时在系统中建立 FTP 目录。以下是在配置 vsftpd 时常用的文件和目录清单：

1) /etc/rc.d/init.d/vsftpd

/sbin/service 命令所需的初始化脚本(initscript)，用于启动、中止或重启 vsftpd。
(更多信息请参考 6.3 启动和中止 vsftpd)。

2) /etc/pam.d/vsftpd

vsftpd 即插即用认证模块 (Pluggable Authentication Modules(PAM)) 的配置文件。这个文件定义用户在登录 FTP 服务器时所需要通过的资格印证过程。

3) /etc/vsftpd/vsftpd.conf

vsftpd 配置文件，该文件包含的选项清单请参考章节 6.5 vsftpd 配置选项。

4) /etc/vsftpd/ftpusers

禁止登录 vsftpd 的用户清单。其缺省状态包括 root、bin、daemon users、among others。

5) /etc/vsftpd/user_list

这一文件既可被配置成禁止登录的用户列表，也可被配置为允许登录的用户列表，这取决于/etc/vsftpd/vsftpd.conf 中的 userlist_deny 参数被设置为 YES(缺省状态)还是 NO；如果本文件被用于配置允许登录的用户列表，则其中的用户名不能出现在/etc/vsftpd/ftpusers 文件中。

6) /var/ftp/directory

这一目录存放 vsftpd 所需的文件，且包括为匿名用户使用的“/var/ftp/pub/directory”。他们对任何用户均可读，但仅对 root 用户是可写的。

6.3 启动和中止 vsftpd

/sbin/service 命令可调用安装的/etc/rc.d/init.d/vsftpd 脚本以实现相应的功能。

启动 ftp 服务器，请以 root 用户身份键入：

```
/sbin/service vsftpd start
```

中止 ftp 服务器，请以 root 用户身份键入：

```
/sbin/service vsftpd stop
```

重启是中止并接着启动 **vsftpd** 的一种简便方法,在希望使被修改了的配置生效时,我们常常用到这一命令;重启 **ftp** 服务器,请以 **root** 用户身份键入:

```
/sbin/service vsftpd restart
```

有条件重启 **condrestart(conditional restart)**仅重启 **vsftpd** 正在运行的部分;这在编写脚本时很有用,因为它不会启动那些没有运行的程序。

有条件重启 **ftp** 服务器,请以 **root** 用户身份键入:

```
/sbin/service vsftpd condrestart
```

在缺省状态下, **vsftpd** 服务在系统启动时并不自动启用,如需实现这一功能,可使用一些初始化工具,如: **/sbin/chkconfig**, **/sbin/ntsysv**, 或者服务配置程序。

6.4 启用 **vsftpd** 的多个副本

有时,一台计算机会被用来服务于多个 **FTP** 域,即 **multihoming** 技术。实现 **vsftpd** 多重应用的一个方法是:运行各自拥有自己的配置文件的多个程序副本。

首先,将所有相关 **IP** 地址分配给系统中的网络设备或网络设备别名。

然后, **FTP** 域的 **DNS** 服务器必须被配置成正确的机器,如 **DNS** 服务器运行的是相应的 **linux** 系统,可使用 **BIND** 域名服务配置。

Vsftpd 根据不同的 **IP** 地址来响应需求,因此必须运行多个副本的程序。第一个副本执行 **vsftpd initscript**, 且使用标准的配置文件 **/etc/vsftpd/vsftpd.conf**。其余 **FTP** 站点的配置文件位于 **/etc/vsftpd/** 目录下,并具有不同的名字而作为其唯一标识,如: **/etc/vsftpd/vsftpd-site-2.conf** 等。只有 **root** 用户才能读写这些配置文件。每个配置文件中 **IPv4** 网络监听语句必须是不同的:

```
listen_address=N.N.N.N
```

N.N.N.N 是相应的 **FTP** 站点所服务的 **IP** 地址。如该 **FTP** 站点使用 **IPv6**, 请用 **listen_address6** 替代上面语句的相应部分。

当系统运行着具有不同配置文件的多个服务时, **vsftpd** 程序必须在 **root** 命令

提示符下启动，命令语句如下：

```
vsftpd /etc/vsftpd/<configuration-file> &
```

其中，<configuration-file>代表不同配置文件的名字，如“/etc/vsftpd/vsftpd-site-2.conf”。

还有其它一些指令，如：

- 1) anon_root
- 2) local_root
- 3) vsftpd_log_file
- 4) xferlog_file

vsftpd 配置文件中用到指令的详细列举请参考 6.5 vsftpd 配置选项。

为将其它服务器也配置成在系统引导时自动启动，请将上面的命令加在 /etc/rc.local 文件的末尾。

6.5 vsftpd 配置选项

尽管 vsftpd 也许并不像其它一些流行的 FTP 服务那样提供用户自定义级别的设置，但它的各种选项配置参数足以实现绝大多数管理者的需求。实际上，它不强调过度个性化的特性恰恰很好地限制了配置和程序上的错误。

Vsftpd 的所有配置均由配置文件/etc/vsftpd/vsftpd.conf 来管理，文件中每一条指令占有一行并遵循下面的格式：

```
<directive>=<value>
```

<directive>与<value>之间的等号两侧没有空格！

注释行须以井号(#)开头，后面的文字将被程序忽略。

以下是配置文件/etc/vsftpd/vsftpd.conf 中一些比较重要的指令，在文件中，如果没有明确定义，则所有指令均按缺省值设定。

6.5.1 程序选项

以下为控制 vsftpd 程序的各种操作的指令列表：

- 1) Listen

当其被激活时，vsftpd 以各自独立的模式运行。Linux 系统设置它为 YES；

该指令不能和 `listen_ipv6` 指令一起使用。缺省值为 NO。

2) `listen_ipv6`

当其被激活时, `vsftpd` 以各自独立的模式运行, 但仅监听 IPv6 接口; 该指令不能和 `listen` 指令一起使用。缺省值为 NO。

3) `session_support`

当其被激活时, `vsftpd` 将通过 Pluggable Authentication Modules (PAM)为每一个用户维持其登录时的 sessions。如果不需要 session 日志, 关闭该选项可使 `vsftpd` 以较少的程序和较低的优先级运行。缺省值为 YES。

6.5.2 登录选项和访问控制

以下是控制登录操作和存取访问的指令列表:

1) `anonymous_enable`

当其被激活时, `ftp` 允许匿名用户访问。缺省值为 YES。

2) `banned_email_file`

如被设置为 YES, 则该指令禁止包含未知密码的邮件进入服务器。缺省值为 `/etc/vsftpd/banned_emails`。

3) `banner_file`

当与服务器建立连接时, 选出所有包含有指定文本的文件, 这一选项将覆盖 `ftpd_banner` 指令中指定的任何文本。该指令没有缺省值。

4) `cmds_allowed`

选出服务器允许的那些 FTP 命令 (以逗号分隔), 其余均被拒绝。该指令没有缺省值。

5) `deny_email_enable`

当其被激活时, 任何匿名用户如使用了在 `/etc/vsftpd/banned_emails` 中列出的邮件密码, 将被阻止访问服务器; 该指令一般和 `banned_email_file` 配合使用。缺省值为 NO。

6) `ftpd_banner`

当其被激活时, 指令中指定的字符串将在与服务器建立连接时显示, 这样选项会被 `banner_file` 指令覆盖。缺省时, `vsftpd` 将显示它的标准 banner。

7) `local_enable`

当其被激活时，本地用户将被允许访问系统。缺省值为 NO。

8) pam_service_name

指定 vsftpd PAM 服务器的名字。缺省值为 ftp，但 Linux 系统为其设置的名字为 vsftpd。

9) tcp_wrappers

当其被激活时，TCP 嵌套将被应用于对服务器的访问；如果 FTP 服务被配置了多个 IP 地址，可利用 VSFTPD_LOAD_CONF 根据用户需求的不同 IP 地址装载相应的配置文件。缺省值为 NO，但 Linux 系统将其设置为 YES。

10) userlist_deny

当其与 userlist_enable 指令配合使用且设置为 NO，除在 userlist_file 中列出的用户，其余本地用户均将被拒绝访问；因为用户在被要求输入密码以前既被拒绝，将该指令设置为 NO 可阻止用户在网络上提交未加密的密码。缺省值为 YES。

11) userlist_enable

当其被激活时，在 userlist_file 中列出的用户均被拒绝访问。因为用户在被要求输入密码以前既被拒绝，将该指令设置为 NO 可阻止用户在网络上提交未加密的密码。缺省值为 NO，但 Linux 系统将其设置为 YES。

12) userlist_file

当 userlist_enable 指令被激活时，列举 vsftpd 需要的文件清单。缺省值为 /etc/vsftpd/user_list。

6.5.3 匿名用户选项

以下列出的是控制匿名用户访问的一些指令，anonymous_enable 必须被设置为 YES 时才能使用它们。

1) anon_mkdir_write_enable

当与 write_enable 连用并激活时，匿名用户可以在具有写权限的目录下新建目录。缺省值为 NO。

2) anon_root

指定当匿名用户登录后，vsftpd 将转向的目录。该指令没有缺省值。

3) anon_upload_enable

当与 write_enable 连用并激活时，匿名用户可以在具有写权限的目录下上传

文件。缺省值为 NO。

4) anon_world_readable_only

当其被激活时，匿名用户仅允许下载那些对所有用户均可读的文件。缺省值为 YES。

5) ftp_username

指定为匿名 FTP 用户使用的本地用户帐号(在/etc/passwd 中列出)；
/etc/passwd 中指定的用户个人目录即为匿名用户登录的根目录。缺省值为 ftp。

6) no_anon_password

当其被激活时，匿名用户无需输入密码。缺省值为 NO。

6.5.4 本地用户选项

以下指令用于定义本地用户对服务器的访问，为使用它们，local_enable 必须被设置为 YES。

1) chmod_enable

当其被激活时，允许本地用户使用 FTP 命令 SITE CHMOD，用户可通过该命令改变文件的“许可”属性。缺省值为 YES。

2) chroot_list_enable

当其被激活时，chroot_list_file 中列举的本地用户将在登陆后被置入 chroot jail。当本选项被激活并同 chroot_local_user 连用时，chroot_list_file 中指定的本地用户在登陆后将不被置入 chroot jail。缺省值为 NO。

3) chroot_list_file

当 chroot_list_enable 被设置为“YES”时，指定包含本地用户清单的文件。
缺省值为/etc/vsftpd/chroot_list。

4) chroot_local_user

当其被激活时，本地用户登录后将转向它的个人用户目录。缺省值为 NO。
使用这一配置将带来一些安全问题，尤其是对于具有上传权限的用户，因此，我们不推荐使用。

5) guest_enable

当其被激活时，所有在 guest_username 中设置的非匿名本地用户将以客人(guest)身份登录。缺省值为 NO。

6) guest_username

指定客人用户身份被映射的用户名。缺省值为 ftp。

7) local_root

指定当本地用户登录时 vsftpd 要转向的目录。该指令没有缺省值。

8) passwd_chroot_enable

当其被激活并与 chroot_local_user 连用时，vsftpd 根据位于该本地用户个人目录中的/etc/passwd 中是否出现./来改变本地用户的路径。缺省值为 NO。

9) user_config_dir

将路径指到包含系统本地用户配置文件的路径，这个文件包含该用户的特定设置，在该用户配置文件中的所有指令将重写“/etc/vsftpd/vsftpd.conf”中的设定。该指令没有缺省值。

6.5.5 目录选项

以下是一些影响目录的指令：

1) dirlist_enable

当其被激活时，允许用户查看目录列表。缺省值为 YES。

2) dirmessage_enable

其激活时，当用户访问一个带有消息文件的目录时会显示一条消息，这条消息放在被访问的目录中，其文件名在 message_file 中指定，一般缺省为.message。缺省值为 NO，但 Linux 系统将其设置为 YES。

3) force_dot_files

当其被激活时，以圆点号(.)起始的文件会被列在目录列表里，但. 和.. 文件除外。缺省值为 NO。

4) hide_ids

当其被激活时，所有目录列表用 ftp 代表用户，group 代表文件。缺省值为 NO。

5) message_file

当使用 dirmessage_enable 时用于指定消息文件的名称。缺省值为.message。

6) text_userdb_names

当其被激活时，测试用户名和组名将用于替代 UID 和 GID 登录；该指令的激活可能影响服务器的速度。缺省值为 NO。

7) use_localtime

当其被激活时，目录列表显示当地时间而不是 GMT（格林威治标准时间）。缺省值为 NO。

6.5.6 文件传输选项

下列指令影响文件的操作：

1) download_enable

当其被激活时，允许下载文件。缺省值为 YES。

2) chown_uploads

当其被激活时，所有由匿名用户上传的文件为 chown_username 中指定的用户所有。缺省值为 NO。

3) chown_username

当 chown_uploads 启用时，指定匿名上传文件的所有者。缺省值为 root。

4) write_enable

当其被激活时，一些用于改变文件系统的 FTP 命令允许被使用，如：DELE、RNFR 和 STOR 等，缺省值为 NO。

6.5.7 日志选项

以下指令控制 vsftpd 的日志操作：

1) dual_log_enable

当其被激活并与 xferlog_enable 连用时，vsftpd 将同时写两个文件：一个是 wu-ftp-compatible 日志，写入在 xferlog_file 指令中指定的文件(缺省为 /var/log/xferlog)；一个是标准 vsftpd 日志文件，写入在 vsftpd_log_file 指令中指定的文件(缺省为 /var/log/vsftpd.log)。缺省值为 NO。

2) log_ftp_protocol

当其被激活并与 xferlog_enable 连用，且 xferlog_std_format 被设置为 NO 时，所有的 ftp 命令和响应均被记录，这一指令对于调试非常有用。缺省值为 NO。

3) syslog_enable

当其被激活并与 xferlog_enable 连用，所有以通常方式写入在 vsftpd_log_file 中指定的标准 vsftpd 日志文件(缺省为 /var/log/vsftpd.log)的内容将被发送给系统日志管理者而不使用 FTPD（文件传输服务器软件）工具。缺省值为 NO。

4) vsftpd_log_file

指定 vsftpd 日志文件。如果这个文件被使用，则 xferlog_enable 必须被激活，且 xferlog_std_format 必须被设置为 NO；或者，如果 xferlog_std_format 设置为 YES，而 dual_log_enable 是激活的。但需要格外注意的是：如果 syslog_enable 的设置为 YES，那么系统日志将被启用而用以取代了使用本条指令指定的文件。缺省值为/var/log/vsftpd.log。

5) xferlog_enable

当其被激活时，vsftpd 日志连接(仅针对 vsftpd 格式)和日志文件的传输信息由 vsftpd_log_file 指定(缺省为/var/log/vsftpd.log)。若 xferlog_std_format 为 YES，则只记录文件传输信息而不记录之间的连接，而且将使用 xferlog_file 中指定的文件(缺省为/var/log/xferlog)。需要格外注意的是：如果 dual_log_enable 设置为 YES，日志文件和格式均被使用。缺省值为 NO，但 Linux 系统将其设置为 YES。

6) xferlog_file

指定 wu-ftp-compatbile 日志文件。使用这一指令，xferlog_enable 需激活，且 xferlog_std_format 设置为 YES。当 dual_log_enable 设置为 YES 时，同样可以使用。缺省值为/var/log/xferlog。

7) xferlog_std_format

当其被激活并与 xferlog_enable 连用时，只有 wu-ftp-compatbile 的传输日志被写入由 xferlog_file 指定的文件(缺省为/var/log/xferlog)。需要格外注意的是：这个文件只记录文件的传输信息而不记录与服务器的连接信息。缺省值为 NO，但 Linux 系统将其设置为 YES。

为同旧的 wu-ftp FTP 服务器编写的日志文件兼容，linux 系统将 xferlog_std_format 设置为 YES，这种设置表明同服务器间的连接不被记录。为在保留 wu-ftp-compatbile 文件传输日志的同时记录同服务器的连接信息，可设置 dual_log_enable 为 YES。如无需保留 wu-ftp-compatbile 文件传输日志，可设置 xferlog_std_format 为 NO，并用#将该行注释掉或彻底删除。

6.5.8 网络选项

以下指令将应用于 vsftpd 同网络之间的相互作用：

1) accept_timeout

指定用户以被动方式建立连接的最长时间（以“秒”计）。缺省值为 60。

2) anon_max_rate

指定匿名用户的最大数据传输速率（以“字节/秒”计算）。缺省值为 0，表示不限制速率。

3) connect_from_port_20

当其被激活时，在主动数据传输方式下 vsftpd 以足够的优先权运行并开放端口 20。关闭此选项将允许 vsftpd 运行于较低的优先级，但可能会和一些 FTP 客户端不兼容。缺省值为 NO，但 Linux 系统将其设置为 YES。

4) connect_timeout

指定用户以主动方式响应连接的最长时间（以“秒”计）。缺省值为 60。

5) data_connection_timeout

指定数据传输允许被拖延的最长时间（以“秒”计），一旦超出，同远程客户端的连接将被切断。缺省值为 300。

6) ftp_data_port

当 connect_from_port_20 设置为 YES 时，指定主动数据连接的端口。缺省值为 20。

7) idle_session_timeout

指定远程客户端命令之间的最长时间间隔（以“秒”计），一旦超出，同远程客户端的连接将被切断。缺省值为 300。

8) listen_address

指定 vsftpd 监听网络连接的 IP 地址。该指令没有缺省值。

9) listen_address6

当 listen_ipv6 设置为 YES 时，用于指定 vsftpd 监听的 IPv6 地址。该指令没有缺省值。

10) listen_port

指定 vsftpd 监听网络连接的端口。缺省值为 21。

11) local_max_rate

指定登录的本地用户传输数据的最大速率（以“字节/秒”计算）。缺省值为 0，表示不限制速率。

12) max_clients

当客户端均以独立模式运行时，指定允许同时连接的最大数量；如有超出数

量的用户试图登录，则系统会报错。缺省值为 0，表示没有限制。

13) `max_per_ip`

指定允许从同一 IP 地址登录的用户的最大数量。缺省值为 0，表示没有限制。

14) `pasv_address`

为在 NAT (Network Address Translation) 防火墙后面的服务器指定公用 IP 地址，这使 vsftpd 可以为被动模式的连接分发正确的回复地址。该指令没有缺省值。

15) `pasv_enable`

当其被激活时，允许被动连接方式。缺省值为 YES。

16) `pasv_max_port`

指定被动模式 FTP 客户端可使用的最高端口号，这种设置用来限制端口的范围以方便建立防火墙。缺省值为 0，表示没有限制，最大不能超过 65535。

17) `pasv_min_port`

指定被动模式 FTP 客户端可使用的最低端口号，这种设置用来限制端口的范围以方便建立防火墙。缺省值为 0，表示没有限制，最低不能低于 1024。

18) `pasv_promiscuous`

当其被激活时，不检查数据连接是否来源于同一 IP 地址，这一设置仅应用于某些特定类型的通道。缺省值为 NO。

除非特别需要，请不要激活此选项，因为它在控制初始化数据连接的时候忽略了一个重要的安全特征：既检查其来源是否为同一 IP 地址。

19) `port_enable`

当其被激活时，允许主动连接方式。缺省值为 YES。

6.6 其它资源

为获取更多关于 vsftpd 的信息，可寻求如下资源。

6.6.1 安装的文档

`/usr/share/doc/vsftpd-<version-number>/` 目录，其中 `<version-number>` 为所安装 vsftpd 包的版本号。

这个目录中的 README 文件包含本软件的基础信息，TUNING 文件包含一

些基础的操作技巧，SECURITY/目录中包含了 vsftpd 采用的安全模式。

vsftpd man pages — 其中包含大量关于程序和配置文件的操作信息，以下列举一些相对重要的部分：

1) 服务器应用

man vsftpd — 介绍 vsftpd 所用到的命令选项。

2) 配置文件

man vsftpd.conf — 包含 vsftpd 配置文件需要的选项。

man 5 hosts_access — 介绍 TCP 嵌套配置文件中用到的格式和选项：如 hosts.allow 和 hosts.deny。

6.6.2 有用的网址

<http://vsftpd.beasts.org/> — vsftpd 工程网页，这是一个获取最新文档和同软件作者沟通的绝佳去处。

<http://slacksite.com/other/ftp.html> — 这里对主动和被动 FTP 模式之间的区别做了简明的解释。

<http://war.jgaa.com/ftp/?cmd=rfc> — 这里有同 FTP 协议相关的综合列表。

7 SQUID 配置和使用

Linux 与 Squid 的组合做为代理服务器，性能很好，足够几百人的小型局域网使用。下面介绍一些 Squid 的安装及使用技巧。

7.1 Squid 简介

Squid 是一个缓存 internet 数据的软件，它接收用户的下载申请，并自动处理所下载的数据。也就是说，当一个用户想要下载一个主页时，它向 Squid 发出一个申请，要 Squid 替它下载，然后 Squid 连接所申请网站并请求该主页，接着把该主页传给用户同时保留一个备份，当别的用户申请同样的页面时，Squid 把保存的备份立即传给用户，使用户觉得速度相当快。目前，Squid 可以代理 HTTP、FTP、GOPHER、SSL 和 WAIS 协议，暂不能代理 POP、NNTP 等协议。

但让 Squid 缓存任何数据也是不好的，比如信用卡帐号、可以远方执行的 scripts、经常变换的主页等是不合适也是不安全的。Squid 可以进行自动处理，您也可以根据自己的需要设置 Squid，使之过滤掉您不想要的东西。

Squid 可以工作在很多的操作系统中，如 AIX、Digital Unix、FreeBSD、HP-UX、Irix、Linux、NetBSD、Nextstep、SCO、Solaris、OS/2 等，也有不少人在其他操作系统中重新编译过 Squid。

Squid 对硬件的要求是内存大于 128M，硬盘转速越快越好，最好使用服务器专用 SCSI 硬盘，处理器 400MH 以上即可。

7.2 Squid 的编译和运行

中标麒麟服务器操作系统中有已经编译好的 Squid，安装它即可。如果手头没有现成的编译好的 Squid 或想使用最新的版本，可去 <ftp://squid.nlanr.net> 下载一份，自己编译。

Squid 的编译是非常简单的，因为它基本上是自己配置自己。最容易出现的问题是您的系统上没有合适的编译器，这可以通过安装相应的编译器解决。如果出现其他问题，您可以问一下有经验的用户或到相应的邮件列表寻找帮助。

编译 Squid 之前，最好建一个专门运行 Squid 的用户和组。例如可以在自己的服务器上建了一个名为 squid 的用户和组，用户目录设为 /usr/local/squid。然后 su 到用户 squid 并从 [squid.nlanr.net](ftp://squid.nlanr.net) 下载 Squid 的源文件到目录 /usr/local/squid/src

中，用如下命令进行解压：

```
$tar xzf squid-2.0.RELEASE-src.tar.gz $cd /usr/local/squid/src/ squid-*.*.RELEASE /  
$./configure  
$make  
$make install
```

第一个命令在目录/usr/local/squid/src中产生一个新的子目录/squid-*.*.RELEASE/。命令./configure会自动查询您的系统配置情况以及您系统中使用的头文件。不加参数的./configure会把Squid安装在目录/usr/local/squid中，如果您想使用其他目录，用如下命令：

```
./configure --prefix=/some/other/directory
```

这会把Squid安装在目录/some/other/directory中。make命令编译Squid，make install命令安装Squid。不出意外的话，目录/usr/local/squid中会出现如下目录：

1) /bin

含有Squid可执行程序，包括Squid本身，ftpget等。

2) /cache

包含Squid缓存的数据，其中包含象/00/ /01/ /02/以及/03/这样的目录，这些目录中还有子目录，因为在多个目录比在一个目录成千上万的文件中寻找一个文件更容易，速度更快。

3) /etc

包含Squid的唯一的配置文件squid.conf。

4) /logs

包含Squid的日志。

5) /src（由自己创建）。

7.3 squid.conf 文件的配置

中标麒麟高级服务器操作系统自带安装Squid的目录为/etc/squid/，在该目录中squid.conf为配置文件，文件中对每一个选项都有详细的说明，用户可以通过修改该文件以满足不同的需要。

总的来说，有如下两个重要选项：

1) http_port

设定 Squid 监听的端口，建议您最好设一个比较好记的端口号，以便在进行客户机配置时容易记住。例如端口号设置为 8080。缺省为 3128。

2) cache_dir

设定缓存的位置、大小。一般形式如下 `cache_dir ufs /var/spool/squid 100 16 256`。/var/spool/squid 代表缓存的位置；100 代表缓存最大为 100M；16 和 256 代表一级和二级目录数。

7.4 运行 Squid

首先以 root 身份登录。

如果您想前台执行 Squid，接着执行命令：

```
#/usr/sbin/squid -NCd1
```

该命令正式启动 Squid。如果一切正常，您会看到一行输出：

```
Ready to serve requests.
```

如果想后台运行 Squid，把它做为一个精灵进程，执行命令：

```
#/usr/sbin/squid&
```

观察 Squid 是否运行使用命令：

```
# squid -k check
```

输出会告诉您 Squid 的当前状态。

8 LDAP 配置和使用

8.1 什么是 LDAP?

LDAP(Lightweight Directory Access Protocol, 轻量目录访问协议)是被提议在网络或 Internet 上全球或本地目录服务的开放标准,这种情况下目录很像电话簿。LDAP 可以处理别的信息,但是目前它被典型的用来联系电话号码和邮件地址的名字。设计 LDAP 目录支持大量的查询,但是存储在目录中的数据并不经常改变。

LDAP 比电话簿更加有用,因为 LDAP 的设计是有意支持在 Internet 上 LDAP 服务器的传播,很像域名服务(DNS)。DNS 服务器在基于完全合法的域名或者来自域要求的服务类型的情况下,有助于连接彼此的计算机,例如邮件交换。如果没有 DNS 服务器,主机名不能被翻译成 IP 地址,而 IP 地址正是 TCP/IP 通信所需要的。将来,LDAP 可以提供相同类型的全球访问给许多的目录信息类型。目前,LDAP 作为目录服务更普遍的使用在一个单一的大组织,例如学校或者公司。

LDAP 是客户—服务器系统。一个 LDAP 客户连接到一个 LDAP 服务器,要么查询服务器得到信息,要么提供需要输入到目录中的信息。服务器要么回答查询,提交查询到另一个 LDAP 服务器,要么接受输入到目录的信息,这些都是基于用户的允许。

LDAP 有时被当作 X.500。X.500 是目录的国际化标准,但是它复杂,需要很多处理资源和完全的 OSI 堆存储器。LDAP 和它相反,可以很容易地运行在 PC 或者 TCP/IP 上。LDAP 可以访问 X.500 目录但是不支持所有的 X.500 的性能。

本节将讨论 LDAP 2.0 的配置和使用,它是对于 LDAPv2 和 LDAPv3 协议的一个开放源码执行。

LDAP 使用户可以跨网络存取联络信息。用户之间可以分享相同的联络信息。典型的 LDAP 应用就是公司中所有雇员共同使用的通讯簿,所有的雇员都可以访问它。

例如,要在 Thunderbird 邮件和日历系统中配置 LDAP 服务器,请执行以下操作:

在 Thunderbird 主窗口中选择上方工具栏【工具】下【通讯录】选项，弹出【通讯录】页面。如图 8-1 的窗口显示界面。

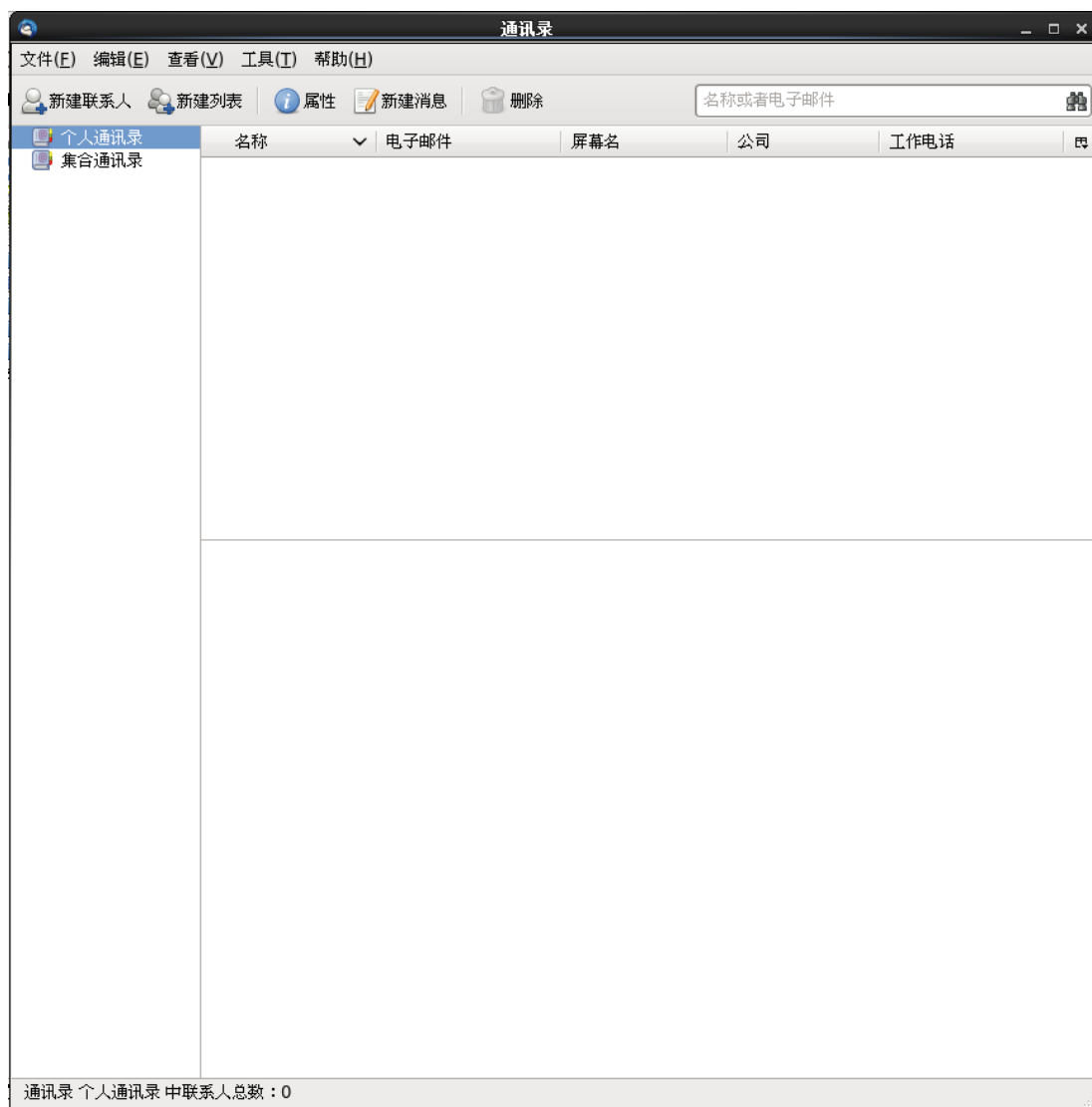
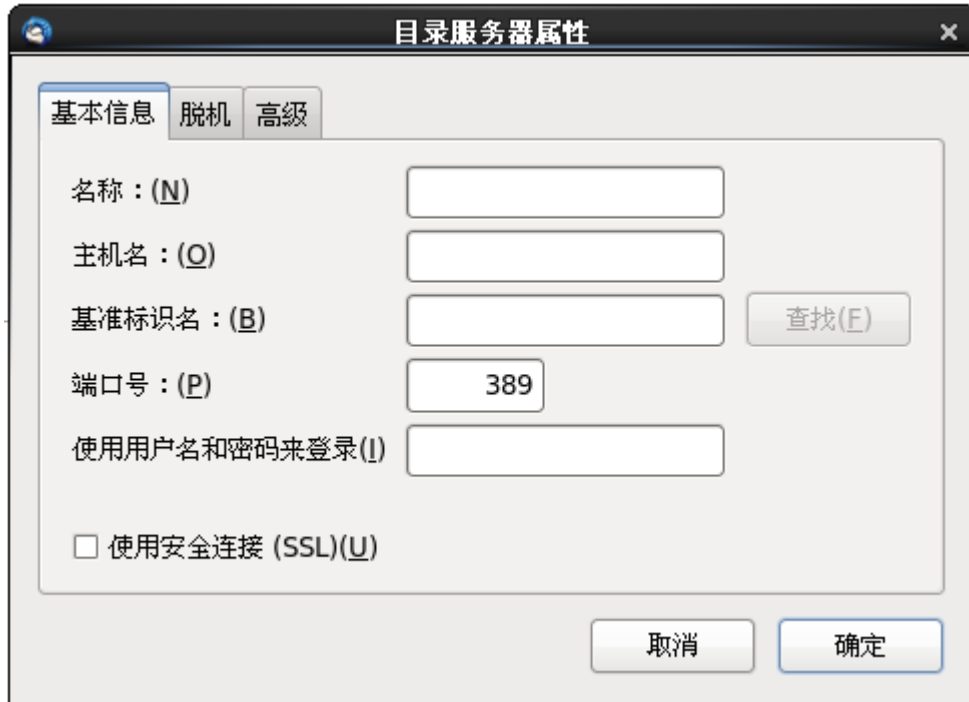


图 8-1 通讯录

点击左上方工具栏【文件】下【新建】→【LDAP 目录...】，即打开了如图 8-2 的【目录服务器属性】窗口。



The image shows a dialog box titled "目录服务器属性" (Directory Server Properties). It has three tabs: "基本信息" (Basic Information), "脱机" (Offline), and "高级" (Advanced). The "基本信息" tab is selected. Inside the tab, there are several input fields and a checkbox:

- 名称 : (N) [Input field]
- 主机名 : (O) [Input field]
- 基准标识名 : (B) [Input field] [Find (F) button]
- 端口号 : (P) [Input field with value 389]
- 使用用户名和密码来登录 (I) [Input field]
- ☐ 使用安全连接 (SSL)(U)

At the bottom right, there are two buttons: "取消" (Cancel) and "确定" (OK).

图 8-2 目录服务器属性

配置选项如下所示：

1) **【名称】**

键入 LDAP 服务器的名称，此名称会显示在 Thunderbird 界面上。

2) **【主机名】**

输入联系人信息所在的 LDAP 服务器的主机名或 IP 地址。

3) **【基准标识名】**

输入联系人信息所在 LDAP 服务器的基准标识名。

4) **【端口】**

在此字段中输入 Thunderbird 用于连接 LDAP 服务器的端口号。此字段的通用值是 389。

5) **【使用用户名和密码来登录】**

如使用 LDAP 地址簿需要登录名和密码，则在这里进行填写。

6) **【使用安全连接】**

通过勾选该选项前的复选框，选择是否使用 SSL（Secure Sockets Layer，安全套接层）连接 LDAP 服务器。

信息设置完成后，单击**【确定】**按钮，将保存设置并关闭本页面，已经创建的 LDAP 服务器地址簿会显示在左侧栏中。

8.2 LDAP 的优点

使用 LDAP 主要的好处是组织中某些类型的信息合并。例如，所有在组织中的不同的用户列表可以合并成一个 LDAP 目录。任何启动 LDAP 需要这些信息的应用程序都可以查询此目录。需要目录信息的用户也可以使用。而且，由于 LDAP 支持 SSL 和 TLS，敏感数据能够得到很好的保护。

LDAP 同样支持多种存储目录的后台数据库，这使得管理员能够根据服务器要分发信息的类型而灵活开发最适合的数据库。因为 LDAP 还具有一个设计优良的客户端应用程序接口(API)，LDAP 可适用的程序在数量和质量上均有很大的范围空间。

8.2.1 OpenLDAP 2.0 的增强特性

OpenLDAP 2.0 包含很多重要特性：

1) LDAPv3 支持

支持 SASL、TLS 和 SSL；自从 LDAPv2 以后很多的改变，使 LDAP 变的更加安全。

2) Ipv6 支持

支持下一代 IP 协议 (IPV6)。

3) IPC 上的 LDAP

OpenLDAP 可以在特定的系统 (IPC) 中通信而不必通过网络，使它更安全。

4) 更新的 CAPI

改善了程序员连接和使用 LDAP 目录服务的途径。

5) LDIFv1 支持

完全兼容 LDAP 数据交换格式(LDIF)的版本 1。

6) 增强的独立 LDAP 服务器

包括更新的访问控制系统、线程库、工具等。

8.3 LDAP 术语

1) entry

项，是 LDAP 目录中的一个单位；每一个“项”用唯一的 DN (Distinguished Name) 标识。

2) attributes

项有属性(attributes)，它是直接和项联系的信息。例如，组织可能是 LDAP 项，和组织相关的属性可能是他的传真号、地址等等。人也可以是 LDAP 目录中的项，人的普通属性包括他们的电话号码和邮件地址。

某些属性是需要的，有些属性是可选的。objectclass(对象类)用于设置某一项的哪些属性是必须的，哪些属性是可选的。对象类的定义在各种各样的 schema 文件中，它们位于/etc/openldap/schema 目录。更多信息请参考 8.6 /etc/openldap/schema/目录。

3) LDIF

LDAP 数据交换格式(LDIF)是 LDAP 项的 ASCII 文本格式。从 LDAP 服务器输入或输出数据的文件必须是 LDIF 格式。LDIF 项如下所示：

```
[<id>]
dn: <distinguished name>
<attrtype>: <attrvalue>
<attrtype>: <attrvalue>
<attrtype>: <attrvalue>
```

一个项可能包括很多<attrtype>，因此需要<attrvalue>对。空行表示一个项已经结束。



注意：<ATTRTYPE>和<ATTRVALUE>对在它们使用以前必须在 SCHEMA 里定义。不能简单地在 LDIF 文件中定义它们，同样不能希望在 SCHEMA 文件中没有相关的数据的 LDAP 服务器能使用信息。

4) 用<>括起来的是变量

当增加 LDAP 项的时候可以设置。<id>例外，当增加项时，LDAP 工具设置<id>，它通常是一个数字，不需要手动的设置。

8.4 OpenLDAP 守护进程和应用程序

OpenLDAP 库和工具在以下这些包中：

1) Openldap

包括运行 OpenLDAP 服务器和客户端应用的必须程序库。

2) openldap-clients

包括在 LDAP 服务器上浏览和修改目录所需的命令行工具。

3) openldap-servers

4) 包括配置和运行一个 LDAP 服务器所需的服务程序和其它应用。

openldap-servers 包中有两个服务程序：Standalone LDAP Daemon (/usr/sbin/slapd) 和 Standalone LDAP Update Replication Daemon (/usr/sbin/slurpd)。

Slapd 进程是独立的 LDAP 服务，而 slurpd 进程用于在网络上同步两个 LDAP 服务的改变内容。如果网上没有两个以上的 LDAP 服务器，不需要运行 slurpd。

为执行管理功能，openldap-servers 包将安装以下各应用到/usr/sbin/目录：

1) slapadd

从一个 LDIF 文件向 LDAP 目录中增加项；例如，命令/usr/sbin/slapadd -l ldif-input 将读取包含所需新项的 LDIF file: ldif-input。

2) slapcat

从 LDAP 目录中以缺省的格式（Berkeley DB）提取项保存到 LDIF 文件。例如，/usr/sbin/slapcat -l ldif-output 命令向名为 ldif-output 的 LDIF 文件输出所需项。这一命令可用于在 Linux 系统下创建目录。

3) slapcat-gdbm

从 LDAP 目录中以 gdbm 格式提取项保存到 LDIF 文件。Gdbm 是原 LDAP 版本的缺省格式，因此，该命令用于升级原来的 ldap 目录到 2.0 版本。更多信息可参考后面的“从早期版本迁移目录”。

4) slapindex

基于当前内容对 slapd 目录重建索引；当/etc/openldap/slapd.conf 中的索引选项变更时，应该运行这一工具。

5) slappasswd

用/etc/openldap/slapd.conf 配置文件中的 ldapmodify 或 rootpw 值生成加密的用户密码。执行/usr/sbin/slappasswd 来创建密码。



注意：在使用 slapadd、slapcat 或 slapindex 之前确认使用/sbin/service ldap stop 停止了 slapd。否则 LDAP 数据库目录的一致性会受到很大威胁。

参考每个应用程序的帮助页来获得如何使用它们的信息。

openldap-clients 包将一些关于在 LDAP 目录中添加、修改和删除项所需的工具安装在/usr/bin/，这些工具包括：

1) ldapmodify

修改在 LDAP 数据库中的项，通过文件或标准输入接收输入。

2) ldapadd

增加项到目录，通过文件或标准输入接收输入；ldapadd 实际是 ldapmodify -a 的硬链接。

3) ldapdelete

从 LDAP 目录删除项，通过文件或 shell 提示接收输入。

4) ldappasswd

设置 LDAP 用户的密码。

5) ldapsearch

使用 shell 提示在 LDAP 目录中搜索项。

除 ldapsearch 外，您无需一个一个输入命令去对 LDAP 目录做修改，您想要的变化可以保存在一个文件中，然后以这个文件作为输入。文件的格式可参见帮助页。

8.4.1 NSS、PAM 和 LDAP

除 OpenLDAP 包，中标麒麟服务器操作系统还包含了一个 nss_ldap 包，可用以增强 LDAP 同 Linux 和其它 UNIX 环境的整合性能。

nss_ldap 包提供如下：

/lib/libnss_ldap-<glibc-version>.so

/lib/security/pam_ldap.so

libnss_ldap-<glibc-version>.so 模块允许应用程序通过 glibc 的 Nameservice Switch (NSS)接口使用 LDAP 目录来查询用户、组、主机以及其它一些信息，其中的<glibc-version>请用 libnss_ldap 的版本号代替。NSS 允许应用程序通过联合使用 LDAP 和 NIS(Network Information Service)来验证，这样压缩了验证文件。

pam_ldap 模块允许 PAM-aware 应用程序使用 LDAP 目录中的信息验证用户。

PAM-aware 应用程序包括登录控制台、POP 和 IMAP 邮件服务和 Samba 服务。在网络中配置 LDAP 服务器时，所有这些应用都使用一个联合的 ID 和密码，可大大简化管理过程。

8.4.2 PHP4、LDAP 和 Apache HTTP 服务器

中标麒麟高级服务器操作系统包含一个为 PHP 服务器端脚本语言设置的

LDAP 模块。php-ldap 包通过/usr/lib/php4/ldap.so 模块为 PHP4-HTML 嵌入式脚本语言添加 LDAP 支持。这一模块使 PHP4 脚本能够访问 LDAP 目录中的信息。

中标麒麟高级服务器操作系统还为 Apache HTTP 服务装载了 mod_authz_ldap 模块。该模块为主题和客户端 SSL 发行者使用 DN (distinguished name) 的短格式，以确定 LDAP 目录中的用户的 DN。同样，它还可以使用用户在 LDAP 目录项的属性来验证；根据用户（或组）对资源的优先级来决定它们的访问操作；拒绝那些使用过期密码的用户访问。在使用 mod_authz_ldap 模块时，mod_ssl module 是必须的。

 注意：mod_authz_ldap 模块目前还不接受加密的用户密码，这一功能在 mod_auth_ldap 中，但此模块仍处于实验阶段，未包含在本 linux 系统内；详情可见 Apache Software Foundation 的网站 <http://www.apache.org/>

8.4.3 LDAP 客户端应用

LDAP 有用于支持创建和修改目录的图形化客户端界面，但未包含在本系统中。如，可到 <http://www.iit.edu/~gawojar/ldap/> 下载 LDAP Browser/Editor，这就是一个基于 JAVA 的应用工具。

大部分其它的 LDAP 客户端以只读方式进行目录访问，使用它们查询，但不能修改、组织信息如：Sendmail、Mozilla、Gnome Meeting 和 Thunderbird。

8.5 OpenLDAP 配置文件

OpenLDAP 配置文件安装在/etc/openldap/目录。下面是其中一些重要的文件和目录：

1) /etc/openldap/ldap.conf

所有使用 OpenLDAP 库的客户端配置文件，如：ldapsearch、ldapadd、Sendmail、Thunderbird 和 Gnome Meeting。

2) /etc/openldap/slapd.conf

slapd 程序的配置文件，详见“编辑/etc/openldap/slapd.conf”。

3) /etc/openldap/schema/目录

本目录包含 slapd 程序使用的 schema 定义，详见“/etc/openldap/schema/目录”。

 注意：如果安装了 nss_ldap 包，它将会建立一个名叫/etc/ldap.conf 的文件，这一文

件被 nss_ldap 包支持的 PAM 和 NSS 模块所用，详见 8.8 使用 OpenLDAP 配置验证系统。

8.6 /etc/openldap/schema/目录

/etc/openldap/schema/目录包含 LDAP 定义，以前是包含在 slapd.at.conf 和 slapd.oc.conf 文件中。所有的属性语法定义、对象类定义现在都在不同的 schema 文件里。各种 schema 文件的样式在/etc/openldap/slapd.conf 中有参考如下：

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/rfc822-MailMember.schema
include /etc/openldap/schema/autofs.schema
include /etc/openldap/schema/kerberosobject.schema
```



注意：不要修改 OpenLDAP 装载的 schema 文件中定义好的项目。

可以扩展 OpenLDAP 使用的 schema 来支持附加的属性类型和使用缺省 schema 文件做为指南的对象类。为了做这个，在/etc/openldap/schema 目录创建 local.schema 文件。通过在缺省的 include schema 行下添加以下行来在 slapd.conf 中参考新的 schema：

```
include /etc/openldap/schema/local.schema
```

下一步，在 local.schema 文件中开始定义新的属性类型和对象类。很多组织使用来自 schema 文件被缺省安装的已经存在的属性类型和对象类并为了在 local.schema 文件的使用修改它们。当适应组织的当前需求，这可以帮助学习 schema 语法。

扩展 schemas 来符合某些专门的需求是本章的范围。对于写新的 schema 文件请参考 <http://www.openldap.org/doc/admin/schema.html> 获得更多的信息。

8.7 建立 OpenLDAP

本节简要介绍 OpenLDAP 目录的安装和配置，更多信息可参考：

<http://www.openldap.org/doc/admin/quickstart.html> — OpenLDAP 网站上的《Quick-Start Guide》。

创建 LDAP 服务的基础步骤如下：

- 1) 安装 openldap、openldap-servers 和 openldap-clients RPM。
- 2) 编辑/etc/openldap/slapd.conf 文件指定 LDAP 域和服务器。
- 3) 启动 slapd:

```
/sbin/service/ldap start
```

- 4) 配置好 LDAP 后，点击桌面左下角【启动】→【系统】→【管理】→【服务】或在图形终端下运行 system-config-services 配置 LDAP 为在系统引导时自启动。
- 5) 使用 ldapadd 向 LDAP 目录中添加“项”。
- 6) 使用 ldapsearch 查看 slapd 是否能够正确访问信息。
- 7) 至此，LDAP 目录应该可以正确运行，并可被配置激活各种 LDAP 应用。

8.7.1 编辑/etc/openldap/slapd.conf

slapd.conf 文件位于/etc/openldap 下，它包括 slapd LDAP 服务器需要的配置信息。您需要编辑此文件使它成为自己的域和服务器的特定的文件。

为使用 slapd LDAP 服务，需修改它的配置文件/etc/openldap/slapd.conf，以指定正确的域和服务器：

suffix 命令为提供 LDAP 信息的服务器命名，需要由下面的格式被修改而映射真正的域名：

例如：

```
suffix "dc=your-domain,dc=com"
```

改为：

```
suffix "dc=example,dc=com"
```

Rootdn 项是一个用户的 DN (Distinguished Name)，该用户无访问限制，可管理那些设置 LDAP 目录的限制级参数。Rootdn 用户可以被理解为 LDAP 目录的 root 用户。在配置文件中，可以用下面的格式修改 rootdn 命令行：

```
rootdn "cn=root,dc=example,dc=com"
```

在网络上发布一个 LDAP 目录时，修改 `rootpw` 命令行 — 用加密的密码代替缺省值。下面的命令用于生成加密的密码：

```
slappasswd
```

出现提示后，键入想要设定的密码，然后再输一遍确认，程序会在 shell 提示符下输出加密后的结果。

然后，将生成的加密密码复制到 `/etc/openldap/slapd.conf` 的一个 `rootpw` 行，删除注释的“#”。

完成后，会得到类似下面的命令：

```
rootpw {SSHA}vv2y+i6V6esazrlv70xSSnNAJE18bb2u
```

 注意：LDAP 密码，包括 `/etc/openldap/slapd.conf` 中的 `rootpw` 指令，在网络上是不加密传输的，除非 TLS encryption 是激活状态。关于激活 TLS encryption 可参考 `/etc/openldap/slapd.conf` 中的内容以及它的帮助。

如需更额外的安全，可以在向网络发布 LDAP 目录前用#将 `rootpw` 命令行注释掉。

当使用 `/usr/sbin/slapadd` 命令在本地发布 LDAP 目录时，可以不使用 `rootpw` 指令。

 提示：只有 root 用户才能使用 `/usr/sbin/slapadd`，但目录服务器是运行在 `ldap user` 状态的。因此，目录服务器不能用于修改任何由 `slapadd` 生成的文件。在使用 `slapadd` 后，为修订发行的版本，可使用下面的命令：

```
chown -R ldap /var/lib/ldap
```

8.8 使用 OpenLDAP 配置认证系统

本节提供如何使用 OpenLDAP 配置中标麒麟服务器操作系统为认证系统的简明概述。如果不是 OpenLDAP 专家，可能需要比这里提供更多的文件。

1) 安装 LDAP 所必需的包

首先,应该确认合适的包在 LDAP 服务器和 LDAP 客户端都已经安装。LDAP 服务器需要 `openldap-servers` 包。LDAP 客户端需要以下的包: `openldap`、`openldap-clients` 和 `nss_ldap`。

2) 编辑配置文件

a) 服务器端

编辑 `/etc/openldap/slapd.conf` 文件并确认它符合应用的需求。可参考 8.7.1 编辑 `/etc/openldap/slapd.conf` 获取编辑 `slapd.conf` 的方法。

b) 客户端

`/etc/ldap.conf` 和 `/etc/openldap/ldap.conf` 都需要根据需求配置好相应的服务和查询。可运行图形化的【验证配置】工具（在终端下键入命令 `system-config-authentication`），在【用户帐户数据库】标签下选择【LDAP】。

客户端机器的 `/etc/nsswitch.conf` 必须设置为使用 LDAP。运行【验证配置】工具，在【用户帐户数据库】标签下选择【LDAP】。

如手工编辑 `/etc/nsswitch.conf`，在合适的地方插入 `ldap`，例：

```
passwd: files ldap
shadow: files ldap
group: files ldap
```

8.8.1 迁移老的认证信息到 LDAP 格式

`/usr/share/openldap/migration` 目录包括一套迁移老的认证信息到 LDAP 格式的 shell 和 perl 脚本。

 注意：为了使用这些脚本必须安装 perl。

首先，需要修改 `migrate_common.ph` 文件以便它映射正确的域。缺省 DNS 域应该改变成如下的样式：

```
$DEFAULT_MAIL_DOMAIN = "example";
```

缺省 `base` 应改成：

```
$DEFAULT_BASE = "dc=example,dc=com";
```

将用户数据库转化为 LDAP 格式的工作由一组安装在同一目录下的脚本来完成，请参考下面的表格 8-1LDAP 迁移脚本决定使用哪一个来完成您的数据库转换工作，请基于已存在的名字服务来运行合适的脚本。

README 以及/usr/share/openldap/migration/目录下的 migration-tools.txt 提供了更多相关信息。

表格 8-1 LDAP 迁移脚本

已存在的名字服务	LDAP 正在运行吗？	使用此脚本：
/etc flat files	yes	migrate_all_online.sh
/etc flat files	no	migrate_all_offline.sh
NetInfo	yes	migrate_all_netinfo_online.sh
NetInfo	no	migrate_all_netinfo_offline.sh
NIS (YP)	yes	migrate_all_nis_online.sh
NIS (YP)	no	migrate_all_nis_offline.sh

8.9 从早期版本迁移目录

OpenLDAP 使用 Sleepy Cat Software 的 Berkeley DB 系统作为目录的硬盘存储格式，以及 GNU 数据库管理器（gdbm）。因此，在升级早期系统的 LDAP 应用程序前，请执行以下步骤：

升级操作系统前，运行命令：

```
/usr/sbin/slappcat -l ldif-output
```

这一命令输出 LDAP 目录中的“项”到 ldif-output 文件。

升级操作系统，注意不要格式化包含 LDIF 文件的分区。

使用命令重新导入 LDAP 目录至升级后的 Berkeley DB 格式：

```
/usr/sbin/slappadd -l ldif-output
```



注意：如果 LDAP 目录在系统升级前未被导出，使用/usr/sbin/slappcat-gdbm -l ldif-output 命令来导出旧的目录（其中 ldif-output 是输出 LDIF 文件的名字）。这一命令可以生成 slapadd 命令使用的 LDIF 文件。

8.9.1 迁移 1.x 目录

以前，LDAP 服务器使用 `gdbm` 作为目录的硬盘存贮格式。如您的系统是较早版本，必须使用如下命令导出现有的 LDAP 目录：

```
ldbmcat -n <ldif_file>
```

上面的 `<ldif_file>` 是输出文件的名字。然后用下面的命令导入：

```
slapadd -l <ldif_file>
```



注意：只有 `root` 用户才能使用 `/usr/sbin/slapadd`，但目录服务器是运行在 `ldap user` 状态的。因此，目录服务器不能用于修改任何由 `slapadd` 生成的文件。在使用 `slapadd` 后，为修订发行的版本，可使用下面的命令：

```
chown -R ldap /var/lib/ldap
```

8.10 附加资源

关于 LDAP，有很多有用的信息。在开始建立和配置 LDAP 前，请参考这些资源，特别是 OpenLDAP 网址和“LDAP HOWTO”。

8.10.1 已安装的文档

`/usr/share/docs/openldap-<versionnumber>/` 目录 — 包括 README 文档和各种信息。

LDAP 相关帮助页 — 包含各种同 LDAP 相关的应用和配置文件的帮助；以下是一些比较重要的部分：

1) 客户端应用

- a) `man ldapadd` — 介绍如何向 LDAP 目录添加“项”。
- b) `man ldapdelete` — 如何删除 LDAP 目录中的“项”。
- c) `man ldapmodify` — 如何修改 LDAP 目录中的“项”。
- d) `man ldapsearch` — 如何在 LDAP 目录中查找“项”。
- e) `man ldappasswd` — 如何设置和修改 LDAP 用户的密码。

2) 服务器端应用

- a) `man slapd` — LDAP 服务器可用命令行的选项。
- b) `man slurpd` — LDAP 应用服务器可用命令行的选项。

3) 管理应用

- a) `man slapadd` — 向 `slapd` 数据库添加项的工具所使用的命令行选项。
- b) `man slapcat` — 由 `slapd` 数据库导出 LDIF 文件的工具所使用的命令行选项。
- c) `man slapindex` — 基于 `slapd` 数据库生成索引的工具所使用的命令行选项。
- d) `man slappasswd` — 为 LDAP 用户生成密码的工具所使用的命令行选项。

4) 配置文件

- a) `man ldap.conf` — LDAP 客户端配置文件中的格式和选项。
- b) `man slapd.conf` — LDAP 服务器应用 (`slapd` 和 `slurpd`)、以及 LDAP 管理工具 (`slapadd`、`slapcat` 和 `slapindex`) 的配置文件格式和选项。

8.10.2 有用的网址

<http://www.openldap.org> — OpenLDAP 的家，合作开发“应用程序和开发工具的商业化、全特征并且开放的源 LDAP 网址。”

<http://www.padl.com/> — `nss_ldap` 和 `pam_ldap` 的开发者的。

<http://www.kingsmountain.com/ldapRoadmap.shtml> — Jeff Hodges 的 LDAP Road Map 包括到几个有用的 FAQ 的链接和关于 LDAP 协议的有关新闻。

<http://www.newarchitectmag.com/archives/2000/05/wilcox/> — 关于管理 LDAP 组的有用资料。

<http://www.ldapman.org/articles> — LDAP 的好的介绍的文章，包括设计目录树和自定义目录结构的方法。

8.10.3 相关的书

《Implementing LDAP》 — 作者 Mark Wilcox; Wrox Press, Inc 出版。

《Understanding and Deploying LDAP Directory Services》 — 作者 Tim Howes et al.; Macmillan Technical Publishing 出版。

9 SMTP 配置和 IMAP 配置

9.1 SMTP 介绍

SMTP (Simple Mail Transfer Protocol, 简单邮件传输协议) 的目标是可靠高效地传送邮件, 它独立于传送子系统而且仅要求一条可以保证传送数据单元顺序的通道。

SMTP 的一个重要特点是它能够在传送中接力传送邮件, 传送服务提供了进程间通信环境 (IPCE), 此环境可以包括一个网络, 几个网络或一个网络的子网。理解到传送系统 (或 IPCE) 不是一对一的是很重要的。进程可能直接和其它进程通过已知的 IPCE 通信。邮件是一个应用程序或进程间通信。邮件可以通过连接在不同 IPCE 上的进程跨网络进行邮件传送。更特别的是, 邮件可以通过不同网络上的主机接力式传送。

9.2 IMAP 介绍

IMAP (Internet Mail Access Protocol, 交互式邮件存取协议) 是斯坦福大学在 1986 年开发的研发的一种邮件获取协议。它的主要作用是邮件客户端 (例如 MS Outlook Express) 可以通过这种协议从邮件服务器上获取邮件的信息, 下载邮件等。当前的权威定义是 RFC3501。IMAP 协议运行在 TCP/IP 协议之上, 使用的端口是 143。它与 POP3 协议的主要区别是用户可以不用把所有的邮件全部下载, 可以通过客户端直接对服务器上的邮件进行操作。

9.3 SMTP 配置和 IMAP 配置

前提: 雷鸟电子邮件未配置。系统能联网, 且 DNS 配置正确。

点击【启动】→【Internet】→【雷鸟电子邮件】, 或在桌面左下角小图标快捷方式栏中点击【雷鸟电子邮件】图标, 打开雷鸟电子邮件帐户设置页面。

在【你的名字】中输入用于显示给他人的用户名, 在【电子邮件地址】中输入欲配置的电子邮件地址, 在【密码】中输入该电子邮件帐户的密码, 点击右下角【继续】。

经由网络自动检测匹配后, 如找到匹配的 IMAP 及 SMTP 配置, 则会在下方显示【已通过尝试常用服务器名找到如下的设置项】, 并在下面列出 IMAP 及

SMTP 的配置信息，您可以直接点击右下角**【创建账户】**，完成电子邮件账户的创建。或者您也可以点击左下方**【手动配置】**，在弹出的 IMAP 和 SMTP 配置的**【服务器主机名】**、**【端口】**、**【SSL】**和**【身份认证】**选项中进行手动填写配置，点击右下角**【重新测试】**，确认该配置填写无误后，点击**【创建账户】**，即可完成该电子邮件的 IMAP 及 SMTP 配置。之后您便可以进入电子邮件账户中进行收发邮件了。

10 NTP

10.1 在网络上同步日期和时间

在桌面上，点击左下角【启动】→【系统】→【管理】→【日期和时间】，弹出【日期/时间属性】配置页面，勾选【在网络上同步日期和时间】前的复选框，即可对本机日期和时间进行网络同步，如下图所示：

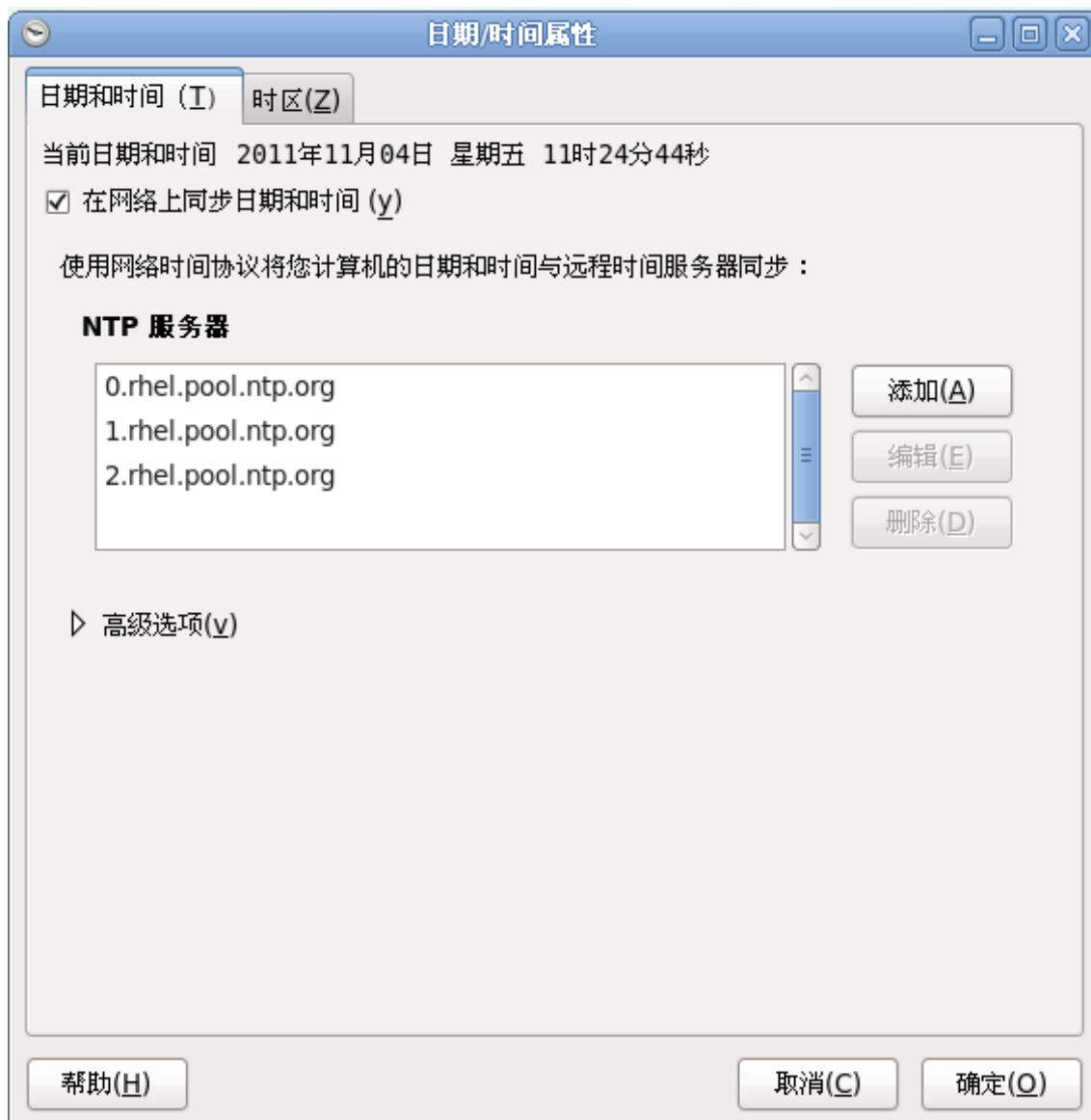


图 10-1 网络时间协议属性

网络时间协议（NTP）进程用远程时间服务器或时间源来同步系统时钟。该程序允许您配置一个 NTP 守护进程来让您的系统时钟与远程服务器同步。要使用该特性，选择【在网络上同步日期和时间】。这会启用【NTP 服务器】列表和

其他选项，如图 10-1 所示，也可以点【**编辑**】来编辑它，或者点【**添加**】来添加一个新时间服务器。点【**确定**】按钮，配置被保存或启动 NTP 守护进程。

点击【**高级选项**】，可以勾选【**加速启动同步**】或者【**使用本地时钟源**】选项。

11 TFTP

11.1 安装 TFTP 服务

插入中标麒麟高级服务器操作系统光盘，root 用户在系统的字符终端下执行命令，进入光盘的挂载路径，用光盘上实际的 vsftpd 包名代替***：

1) 安装软件包：

```
rpm -ivh Server/tftp-server-***.rpm
```

安装完成提示安装成功信息。

2) 查询软件包：

```
rpm -q tftp-server
```

3) 卸载软件包：

```
rpm -e tftp-server --nodeps
```

11.2 TFTP 服务的启动与停止

11.2.1 tftp 服务运行

tftp 服务属于按需服务，系统默认是禁用的，必须在服务器端配置和运行 xinetd 服务以使之可用，方法如下：

1) 方法一：

修改文件/etc/xinetd.d/tftp，把：

```
disable = yes
server_args = -s /tftpboot
```

修改为：

```
disable = no
server_args = -s /tftpboot -c
```

重启 xinetd 服务：

```
/etc/init.d/xinetd restart
```

2) 方法二:

运行下面的命令:

```
chkconfig --level 345 xinetd on  
chkconfig --level 345 tftp on
```

运行下面的命令查看 tftp 服务的状态:

```
chkconfig --list tftp
```

11.2.2 tftp 服务停止

tftp 服务的守护进程名为 in.tftpd, 当超过一定时间(缺省为 15 分钟)没有新的客户端请求时, 该进程将自动停止, 执行下面的操作将立即杀死该进程:

```
pkill in.tftpd
```

tftp 服务属于按需服务, 必须在服务器端配置和重启 xinetd 服务以使之停用, 方法如下:

修改文件/etc/xinetd.d/tftp, 把:

```
disable = no
```

修改为:

```
disable = yes
```

重启 xinetd 服务:

```
/etc/init.d/xinetd restart
```

在客户端使用 tftp 客户端程序连接 tftp 服务并上传文件:

```
tftp server-host-ip  
tftp>put testfile
```

11.3 使用 TFTP 服务上传和下载文件

前提：准备两台互相联网的主机：服务器与客户端，服务器端已经安装了 tftp-server 软件包且 tftp 服务已启动，客户端已经安装了 tftp 软件包，服务器端 IP 表示为 server-host-ip。

操作步骤：

在服务器端修改/tftpboot 目录的权限为所有用户可读可写可执行，这个目录是上传和下载文件的缺省目录：

```
chmod -R 777 /tftpboot
```

在客户端/tmp 目录下创建文件 testfile，用于文件上传测试：

```
touch /tmp/testfile
chmod 777 /tmp/testfile
echo This is the file for upload testing! > /tmp/testfile
```

在客户端打开一个终端，进入/tmp 目录，使用 tftp 客户端程序连接 tftp 服务：

```
tftp server-host-ip
```

在 tftp>命令提示符后输入 verbose 回车，打开 verbose 模式：

在 tftp>命令提示符后输入上传文件 testfile 的命令：

```
tftp>put testfile
```

在 tftp>命令提示符后输入下面的命令下载刚才上传的 testfile 文件到本地 /mnt 目录：

```
tftp>get testfile /mnt/testfile
```

在 tftp>命令提示符后输入 quit 退出连接。

删除测试过程中产生的所有临时文件。

12 SNMP

12.1 snmp 服务的启动与停止

前提：安装 net-snmp-utils 及相关依赖软件包。系统能联网，且 DNS 配置正确。

打开终端，输入命令 `service snmpd start` 启动 snmpd；

终端输入命令 `snmpwalk -v 2c -c public 10.1.9.33 system | more` 查看系统信息；

输入命令 `service snmpd stop` 停止 snmpd；

输入命令 `snmpwalk -v 2c -c public 10.1.9.33 system | more` 查看系统信息。

13 SSH

SSH (Secure Shell, 安全外壳) 是一个协议, 此协议有利于实现使用客户机/服务器架构的两个系统间的安全通信, 并允许用户远程登录服务器主机系统。与其他远程通信协议 (如 FTP 或 Telnet) 不同, SSH 加密登录会话, 导致入侵者难于获取非加密的密码, 而无法连接。

ssh 程序旨在取代老式的、登录远程主机的不安全终端应用, 如 telnet 或 rsh。一个相关的称为 scp 的程序取代了老式程序 (如 rcp), 该程序旨在主机间拷贝文件。因为这些老式应用程序不对客户端和服务端传输的密码进行加密, 只要可能就尽量避免对密码进行加密。使用安全的方法登录到远程系统, 这降低了客户系统和远程主机的风险。

中标麒麟高级服务器操作系统包括通用 OpenSSH 软件包(OpenSSH)以及 OpenSSH 服务器 (openssh-server) 和客户端 (openssh-clients) 软件包。注意, OpenSSH 软件包需要 OpenSSL 软件包(OpenSSL), 其中, OpenSSL 安装了几个重要的加密库, 使 OpenSSH 能提供加密通信。

13.1 SSH 协议

13.1.1 为什么使用 SSH?

入侵者可能使用各种中断、拦截和重新分配网络流量的工具, 企图访问系统。总的来说, 这些威胁可以分类如下:

1) 拦截两系统间的通信

攻击者可能处于通信双方之间网络的某处, 复制在他们之间传递的信息。他可以拦截并保留信息, 或者改变信息并将其发送给预定收件人。

攻击者经常通过使用数据包探测器来完成这种攻击, 数据包探测器是一个相当常见的网络实用工具, 此工具捕捉经过网络的每个数据包, 并分析数据包的内容。

2) 模仿特定主机

配置系统时, 攻击者将其系统伪装成数据传输的预定收件人。如果伪装成功, 用户的系统依然不知道它在与错误的主机通信。

攻击者使用一种被称作 DNS 病毒的技术, 或所谓的 IP 欺骗技术来实现这种

攻击。第一种情况下，入侵者使用破解的 DNS 服务器，将客户端系统指引到恶意复制的主机上。在第二种情况下，入侵者发送虚假的网络数据包，此数据包好像是来自于被信任的主机。

这两种技术都可能拦截敏感信息，若是恶意拦截，将造成灾难性结果。如果将 SSH 用于远程外壳登陆和文件复制，就可以大大减少这些安全威胁。因为 SSH 客户端和服务端使用数字签名来验证他们的身份。此外，所有客户端和服务端之间的通信都被加密。因为使用只有本地和远程系统才知道的密钥对每个包都进行加密，所以企图欺骗任一通信方的做法将失败。

13.1.2 主要功能

SSH 协议提供下列保护措施：

1) 没有人能伪装成预定服务器

初次连接后，客户端可以验证：它正在连接的服务器与他以前连接的服务器是同一台服务器。

2) 没有人可以获取身份认证信息

客户端使用 128 位加密技术将其身份认证信息传输到服务器。

3) 没有人可以拦截通讯

使用 128 位加密技术传输对话期间发送和接收的所有数据，使拦截者难于解密和阅读被拦截的传输。

另外，SSH 协议还提供以下选项：

1) 它提供使用网络上图形化应用的安全手段。

通过采用一种名为 X11 转发的技术，客户端可以从服务器转发 X(X 视窗系统)应用。

2) 它提供了一种保护其他不安全协议的方法。

SSH 协议对它发送和接收的一切都加密。采用一种名为端口转发的技术，一个 SSH 服务器可以作为一个通道来保护其他不安全协议（如 POP），从而提高整个系统和数据的安全性。

3) 它可以用于创建安全通道。

它可以配置 OpenSSH 服务器和客户端，使它们能够创建用于服务器和客户机通信的、类似于虚拟专用网络的通道。

4) 可以支持 Kerberos 身份认证。

它可以配置 OpenSSH 服务器和客户端，使 OpenSSH 服务器和客户端使用 Kerberos 网络身份认证协议的 GSSAPI(通用安全服务的应用程序接口)实现来进行身份认证。

13.1.3 协议版本

SSH 目前存在两种变体：版本 1 和新版本 2。中标麒麟高级服务器操作系统下的 OpenSSH 工具包使用 SSH 版本 2，SSH 版本 2 具有一个增强型的密匙交换算法，不易受到版本 1 中大家都知道的攻击。然而，出于兼容性的原因，OpenSSH 工具包也支持符合版本 1 的连接。

 注意：避免使用 SSH 版本 1。为保证您的连接的最大安全性，建议只要有可能，就尽量使用使用与 SSH 版本 2-兼容的服务器和客户端。

13.1.4 SSH 连接的事件序列

下面一系列事件有助于保护两主机之间 SSH 通信的完整性。

- 1) 采用加密的握手方式，使客户端可以验证它正在与正确的服务器进行通讯。
- 2) 使用对称加密方法对客户端和远程主机连接的传输层进行加密。
- 3) 客户端向服务器认证其自身的身份。
- 4) 远程客户端和远程主机在被加密的连接上进行交互。

13.1.4.1 传输层

传输层的首要任务是在身份认证时和后续通信时保证两主机间安全可靠的通讯。通过对数据的加密和解密，以及通过在发送和接收数据包时对数据包提供完整性保护，传输层可以完成这个任务。传输层也可以提供加快信息传递的压缩技术。

一旦 SSH 客户联系一台服务器，将交换密钥信息，以便这两个系统能够正确构建传输层。在交换密钥时，发生下面的步骤：

- 1) 交换密钥。
- 2) 确定公钥加密算法。
- 3) 确定对称加密算法。
- 4) 确定消息认证的算法。

5) 确定 hash 算法。

密钥交换时，服务器会通过唯一的主机密钥向客户端识别其本身。如果客户端以前从未与此特定服务器通信过，则客户端不知道服务器的主机密钥从而不能建立连接。OpenSSH 通过接受服务器主机密钥来规避这个问题。这是在用户收到通知，且用户接受并验证了新主机钥匙后完成的。假定客户端确实与预定服务器通信，则在随后的连接中，服务器的主机密钥与客户端保存的密钥进行核对。如果将来主机钥匙与客户端保存的密钥不再匹配，用户必须在连接前删除客户端保存的密钥。



警告：始终核实新 SSH 服务器的真实性：

攻击者有可能在初次联系服务器时伪装成 SSH 服务器，因为本地系统并不知道预定服务器和攻击者设置的伪装服务器之间的区别。若要防止这种情况发生，请在第一次连接之前或发生主机密钥不匹配时联系服务器管理员，验证新 SSH 服务器的真实性。

SSH 旨在可与任何一种公钥算法或编码格式一同使用。在初次密钥交换创建了用于交换的 hash 值和共享秘密数值后，两个系统立即开始计算新密钥和算法，对身份认证和未来通过该连接发送的数据进行保护。

当使用给定密钥和算法传送一定量的数据后(准确的数目取决于 SSH 实现)，发生另一次密钥交换，生成另一套 hash 数据和一个新的共享秘密数值。即使攻击者可以决定 hash 和共享密码数值，这个信息也只在有限的时间段内有用。

13.1.4.2 身份认证

一旦传输层构建了在两个系统之间传递信息的安全通道，服务器通知客户端所支持的不同身份认证方法，例如使用私钥编码签名或输入密码。通过使用其中一种被服务器支持的方法，然后客户端向服务器认证其自身的身份。

用户可以配置 SSH 服务器和客户端，使它们允许不同类型的身份认证，这提供给每一方最佳量的控制。基于服务器的安全模型，服务器可以决定支持哪些加密方法，客户端从可用选项中选择尝试身份认证方法的顺序。

13.1.4.3 通道

成功完成 SSH 传输层上的身份认证后，通过一种叫做 multiplexing 的技术打开多个通道。每个通道为不同终端会话被转发的 X11 对话处理通信。

客户端和服务器都可以创建新通道。在连接结束时，每个通道都被指定了一

个不同的号码。当客户端试图打开一个新通道时，客户端发送通道号码和请求。服务器储存此信息并用来指导与那个通道的通信。这种做法使得不同类型的对话不会互相影响，并且当一个对话结束时，可以关闭通道而不中断主要 SSH 连接。

通道同样支持流控制，流控制允许他们以有序的方式发送和接收数据。通过这种方法，直到客户端接收到通道已被打开的消息时，才会发送数据给通道。

根据客户端请求的服务类别和用户连接到网络的方式，客户端和服务器自动协商每个通道的特性。这使用户可以很灵活的处理不同类型的远程连接，而不需要必须改变协议基本结构。

13.2 OpenSSH 配置

为了执行本部分描述的任务，你必须有超级使用者特权。若要获得它们，请通过输入以下内容来作为 root 用户登录：

```
su- 回车
输入根密码
```

13.2.1 配置文件

有两种不同的配置文件：用于客户端程序的配置文件(即 ssh、scp 和 sftp)，和用于服务器的配置文件(sshd 后台程序)。

系统级的 SSH 配置信息储存在/etc/ssh/目录。见表格 13-1 系统级系统级配置文件，该表描述了系统级配置文件的内容。

表格 13-1 系统级

配置文件	描述
/etc/ssh/moduli	包含 Diffie-Hellman 组对不同的 Diffie-Hellman 钥匙交换，这对建构安全传送层非常重要。当在 SSH 对话初期交换钥匙时，创建了一个共享的秘密数值。由双方单独决定此数值用于提供主机身份认证。
/etc/ssh/ssh_config	默认 SSH 客户端配置文件。如果此文件存在，则它被 ~/.ssh/config 重载。
/etc/ssh/sshd_config	Sshd 守护进程配置文件。
/etc/ssh/ssh_host_dsa_key	Sshd 守护进程使用的 DSA 私钥。

/etc/ssh/ssh_host_dsa_key.pub	Sshd 守护进程使用的 DSA 公钥。
/etc/ssh/ssh_host_key	SSH 协议版本 1 的 sshd 守护进程使用的 RSA 私钥。
/etc/ssh/ssh_host_key.pub	SSH 协议版本 1 的 sshd 守护进程使用的 RSA 公钥。
/etc/ssh/ssh_host_rsa_key	SSH 协议版本 2 的 sshd 守护进程使用的 RSA 私钥。
/etc/ssh/ssh_host_rsa_key.pub	SSH 协议版本 2 的 sshd 使用的 RSA 公钥。

特定用户的 SSH 配置信息被储存在 ~/.ssh/ 目录内的用户的主目录中。见表格 13-2 特定用户，该表描述了特定用户配置文件的内容。

表格 13-2 特定用户

配置文件	描述
~/.ssh/authorized_keys	为服务器保持已被认证的公钥列表。当客户端连接到服务器时，服务器通过检查储存在此文件里的签名公钥来对客户端进行身份认证。
~/.ssh/id_dsa	包括用户的 DSA 私钥。
~/.ssh/id_dsa.pub	用户的 DSA 公钥。
~/.ssh/id_rsa	SSH 协议版本 2 的 sshd 使用的 RSA 私钥。
~/.ssh/id_rsa.pub	SSH 协议版本 2 的 ssh 使用的 RSA 公钥。
~/.ssh/identity	SSH 版本 1 的 ssh 使用的 RSA 私钥
~/.ssh/identity.pub	SSH 版本 1 的 ssh 使用的 RSA 公钥。
~/.ssh/known_hosts	包含被用户访问的 SSH 服务器的 DSA 主机钥匙。 这个文件对于确认 SSH 用户正在连接到正确的 SSH 服务器非常重要。

更多关于 SSH 配置文件中各种可用指令的信息，请参考 ssh_config 和 sshd_config 页面。

13.2.2 启动 OpenSSH 服务器

提示：确保安装了相关的文件包：

若要运行 OpenSSH 服务器，您必须安装了 openssh-server 和 openssh 包。

要启动 sshd 守护进程，请在 shell 提示符后输入以下命令：

```
service sshd start
```

要停止运行 sshd 守护进程，请使用以下命令：

```
service sshd stop
```

如果你需要守护进程在计算机启动时自动运行，请输入：

```
chkconfig sshd on
```

这可以启用所有运行级别的服务。注意，如果你重装系统，将会创建一系列新的身份认证密钥。因此，在重装系统前使用任何 OpenSSH 工具连接到系统的用户将会看到以下信息：

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@  WARNING:REMOTE HOST IDENTIFICATION HAS CHANGED!  @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA hostkey has just been changed.
```

为防止这种情况发生，你可以从/etc/ssh/目录备份相关文件（见表格 13-1 系统级），并在重装系统时恢复这些文件。

13.2.3 请求 SSH 远程连接

为使 SSH 真的有效，应禁止使用不安全连接协议。否则，用户密码只在一次会话中使用 SSH 得到保护，当稍后使用 Telnet 登录时就能获取用户密码。有些要禁用的服务包括 telnet、rsh、rlogin 和 vsftpd。

在 shell 提示符后输入以下命令，可以禁用这些服务。

```
chkconfig telnet off
chkconfig rsh off
chkconfig rlogin off
chkconfig vsftpd off
```

13.2.4 使用基于密钥的身份认证

为进一步保护系统安全，你可以取消标准的密码身份认证，实施基于密钥的身份认证。要做到这一点，请在文本编辑器（如 vi 或 nano）中打开

/etc/ssh/sshd_config 配置文件，并按如下方式修改 PasswordAuthentication 选项：

```
PasswordAuthentication no
```

若要使用 ssh、scp 或者 sftp 实现从客户机连接到服务器，请根据以下步骤生成身份认证密钥对。注意，密钥必须是为每个用户独立生成的。

默认情况下，中标麒麟高级服务器操作系统使用 SSH 协议 2 和 RSA 密钥。



提示：不要作为 root 用户生成密钥对

如果您作为 root 用户完成了上述步骤，则只有 root 用户能使用那些密钥。



注意：备份 ~/.ssh/ 目录

如果你重装系统并想保持之前生成的密钥对，请备份 ~/.ssh/ 目录。重装后，把它复制回您的主目录。您系统上的所有用户都可以完成此过程，包括 root 用户。

13.2.4.1 生成密钥对

要生成 SSH 协议版本 2 的 RSA 密钥对，请遵循以下步骤：

通过在 shell 提示符后输入以下内容来生成 RSA 密钥对：

```
ssh-keygen -t rsa
Generating public/private rsa keypair.
Enter file in which to save the key (/home/john/.ssh/id_rsa):
```

按下 Enter 键确认默认新建密钥的默认位置 (~/.ssh/id_rsa)。

输入密码，如需要再次输入以确认此密码。为安全起见，避免使用与你登陆账号所用密码一样的密码。

之后，你会看到与以下内容类似的信息：

```
Your identification has been saved in /home/john/.ssh/id_rsa.
Your public key has been saved in /home/john/.ssh/id_rsa.pub.
The key fingerprint is:
e7:97:c7:e2:0e:f9:0e:fc:c4:d7:cb:e5:31:11:92:14 john@penguin.example.com
The key's randomart image is:
+---[RSA2048]-----+
|  E.  |
|  . .  |
|  o .  |
|  ..  |
```

```
|      S . . |
|      + o o ..|
|    * * + oo|
|    O + . . =|
|    o *   o. |
+-----+
```

改变 ~/.ssh/ 目录的权限:

```
chmod 755 ~/.ssh
```

把 ~/.ssh/id_rsa.pub 的内容复制到你需要连接到的机器的 ~/.ssh/authorized_keys 里, 如果文件已经存在, 就把 ~/.ssh/id_rsa.pub 的内容附件在文件尾。

使用以下命令, 改变 ~/.ssh/authorized_keys 的权限:

```
chmod 644 ~/.ssh/authorized_keys
```

要生成 SSH 协议版本 2 的 RSA 密钥对, 请遵循以下步骤:

通过在 shell 提示符后输入以下内容来生成 RSA 密钥对:

```
ssh-keygen -t dsa
Generating public/private dsa keypair.
Enter file in which to save the key (/home/john/.ssh/id_dsa):
```

按下 Enter 键确认新建密钥的默认位置 (~/.ssh/id_rsa)。

输入密码, 如需要再次输入以确认密码。为安全起见, 避免使用与你登陆账号所用密码一样的密码。

之后, 你会看到与以下内容类似的信息:

```
Your identification has been saved in /home/john/.ssh/id_dsa.
Your public key has been saved in /home/john/.ssh/id_dsa.pub.
The key fingerprint is:
81:a1:91:a8:9f:e8:c5:66:0d:54:f5:90:cc:bc:cc:27john@penguin.example.com
The key's randomart image is:
+--[DSA1024]----+
|      .oo*o.    |
|     ...oBo     |
```

```
| ...+ o. |
| . . E o |
| o..o S |
|.o = . |
|. + |
| .|
| |
+-----+
```

改变~/.ssh/目录的权限:

```
chmod 775 ~/.ssh
```

把~/.ssh/id_.pub 的内容复制到你需要连接到的机器的~/.ssh/authorized_keys 里, 如果文件已经存在, 就把~/.ssh/id_.pub 的内容附件在文件尾。

使用以下命令, 改变~/.ssh/authorized_keys 的权限:

```
chmod 644 ~/.ssh/authorized_keys
```

若要生成 SSH 协议版本 1 的 RSA 密钥对, 请遵循以下步骤:

通过在 shell 提示符后输入以下内容来生成 RSA 密钥对:

```
ssh-keygen -t rsa1
Generating public/private rsa1 keypair.
Enter file in which to save the key (/home/john/.ssh/identity):
```

按下 Enter 键确认新建密钥的默认位置(~/.ssh/id_rsa)。

输入密码, 如需要再次输入以确认密码。为安全起见, 避免使用与你登录账号所用密码一样的密码。

之后, 你会看到与以下内容类似的信息:

```
Your identification has been saved in /home/john/.ssh/identity.
Your public key has been saved in /home/john/.ssh/identity.pub.
The key fingerprint is:
cb:f6:d5:cb:6e:5f:2b:28:ac:17:0c:e4:62:e4:6f:59 john@penguin.example.com
The key's randomart image is:
+--[RSA1 2048]--+
|  |
```

```
| . . |
| o o |
| + o E |
| .o S |
| = + . |
| . = . o . |
| . = o o . .o |
| .oo o=o. |
+-----+
```

改变~/.ssh/目录的权限：

```
chmod 755 ~/.ssh
```

把 ~/.ssh/identity.pub 的内容复制到你 需要连接到的机器的 ~/.ssh/authorized_keys 里，如果文件已经存在，就把 ~/.ssh/identity.pub 的内容附件在文件尾。

使用以下命令，改变 ~/.ssh/authorized_keys 的权限：

```
chmod 644 ~/.ssh/authorized_keys
```

更多关于如何设置系统记住密码的信息，请参考 13.2.4.2 配置 ssh-agent。

 注意：不要共享你的私钥，私钥仅供个人使用，不能提供给任何人，这一点非常重要。

13.2.4.2 配置 ssh-agent

若要储存密码，从而不用在每次初始化与远程机器的连接时都输入密码，你可以使用 ssh-agent 认证代理。如果您正在运行 GNOME，你可以在登录的任何时候设置密码并在对话中牢记它。或者你可以把密码储存为 shell 提示符。

若要在 GNOME 对话时储存密码，请遵循以下步骤，。

确认安装了 openssh-askpass 包。

选择【系统】→【首选项】→【启动应用】将开启【应用程序首选项】，默认情况下，将显示包含可用的开机启动程序列表的选项卡。



图 13-1 开机启动应用程序

点击右边【添加】，然后在命令字段里键入/usr/bin/ssh-add。



图 13-2 添加新应用程序

点击【添加】并确保新增项目旁的复选框被勾选。



图 13-3 启用应用程序

注销后重新登录。将弹出提示您输入密码的对话框。从此开始，ssh、scp 或 sftp 不会再提示你输入密码。

若要将您的密码保存为 shell 提示符，请使用以下命令：

```
ssh-add
Enter passphrase for/home/john/.ssh/id_rsa:
```

当您注销时，您的密码将被消除。每次登陆虚拟控制台或终端窗口时，你都必须执行此命令。

13.3 OpenSSH 客户端

请确保您已安装相关软件包。

若要将客户计算机与 OpenSSH 服务器连接，您必须安装 OpenSSH-Clients 和 Open ssh 软件包。更多关于如何在中标麒麟公司 linux 系统上安装新软件包的信息。

13.3.1 使用 SSH 功能

ssh 允许您登录远程计算机并在远程计算机上执行命令。它是 rlogin、rsh 和 telnet 程序的一个安全替代者。

与 telnet 类似，例如您想登录名为 `penguin.example.com` 的远程计算机，请在命令行提示符中键入如下指令：

```
ssh penguin.example.com
```

这样您就可使用您在本地计算机的用户名登录远程计算机。若您想用别的用户名，请输入与 `ssh username@hostname` 相同格式的指令。例如，以用户名 John 登录，请输入：

```
ssh john@penguin.example.com
```

您初次发起一个连接时，可能会收到一条诸如此类的信息：

```
The authenticity of host 'penguin.example.com' can't be established.  
RSA key fingerprint is 94:68:3a:3a:bc:f3:9a:9b:01:5d:b3:07:38:e2:11:0c.  
Are you sure you want to continue connecting (yes/no)?
```

请输入 `yes` 确认。您会收到一条“服务器已被添加到已知主机列表”的通知，以及要求输入密码的对话框：



注意：更新 SSH 服务器的 host key

若 `ssh` 服务器的主机密钥被更改，客户端会通知用户该连接无法继续，直至从 `~/.ssh/known_hosts` 文件中删除服务器的主机密钥。做法如下，在文档编辑器中打开该文件，删除开头带有远程计算机名的那行。但是，在此之前，请与 `ssh` 服务器的系统管理员联系，以核实服务器不受影响。

在输入密码后，您会收到远程计算机的命令行提示符。

或者，`ssh` 程序也可在未登录命令行提示符的情况下执行远程计算机上的指令。那样做的语法为 `ssh [username@]hostname command`。例如，您若想在 `penguin.example.com` 计算机上执行 `whoami` 指令，请输入：

```
ssh john@penguin.example.com whoami  
john@penguin.example.com's password:  
john
```

您输入正确的密码后，会显示用户名，接着您就会回到本地命令行提示符。

13.3.2 使用 `scp` 实用工具

`Scp` 可用于在安全加密连接的计算机之间传输文件。它的设计与 `rcp` 非常相

似。

将本地文件传送到远程系统，用如下格式的指令：

```
scp localfile username@hostname:remotefile
```

例如，您想将 `taglist.vim` 文件传到名为 `penguin.example.com` 的远程计算机上，在命令行提示符中输入以下指令：

```
scp taglist.vim john@penguin.example.com:~/.vim/plugin/taglist.vim
john@penguin.example.com's password:
taglist.vim      100%   144KB  144.5KB/s   00:00
```

可一次指定多个文件。要将 `.vim/plugin/` 的内容传到远程计算机 `penguin.example.com` 上的同一个目录，请输入以下指令：

```
scp .vim/plugin/* john@penguin.example.com:~/.vim/plugin/
john@penguin.example.com's password:
closetag.vim  100%   13KB   12.6KB/s  00:00
snippetsEmu.vim 100%   33KB   33.1KB/s  00:00
taglist.vim   100%   144KB  144.5KB/s 00:00
```

将远程文件传到本地系统，用如下语法：

```
scp username@hostname:remotefile localfile
```

例如，要从远程计算机上下载 `.vimrc` 文件，请输入：

```
scp john@penguin.example.com:~/.vimrc .vimrc
john@penguin.example.com's password:
.vimrc  100%  2233 2.2KB/s  00:00
```

13.3.3 用 **sftp** 实用工具

Sftp 实用工具可用于打开安全交互式的 **ftp** 会话。其设计与 **ftp** 相似，不同之处在于它使用安全加密连接。

要连接远程系统，请使用如下格式的指令：

```
sftp username@hostname
```

例如，要以用户名 `john` 登录远程计算机 `penguin.example.com`，请输入：

```
sftp john@penguin.example.com
john@penguin.example.com'spassword:
Connected to penguin.example.com.
sftp>
```

在您输入正确的密码后，您会看到一个对话框。Sftp 实用工具接受的一套命令与 ftp 所用的命令相似。（见表格 13-3“可用 sftp 命令选集”）

表格 13-3 可用 sftp 命令

命令	描述
ls [directory]	列出远程目录的内容，若不提供任何目录，则默认使用当前工作目录。
cd directory	将远程工作目录改变到 directory。
mkdir directory	创建一个远程目录 directory。
rmdir path	删除一个远程目录 directory。
put localfile [remotefile]	将本地文件 localfile 传输到远程机器。
get remotefile [localfile]	从远程机器传输 remotefile。

完整的可用命令表，请参见 sftp 手册页。

13.4 不只是安全命令行

安全命令行界面只是使用 ssh 的诸多方式的开始。若具备适当的带宽，x11 会话可被直接导入进整个 ssh 通道。或者，通过使用 tcp/IP 转发，之前系统间不安全的端口连接都可被映射到特定的 ssh 通道。

13.4.1 X11 转发

要打开 SSH 连接下 X11 对话，请使用如下格式的指令：

```
ssh -X username@hostname
```

例如，要以用户名 John 登录远程计算机 penguin.example.com，请输入：

```
ssh -X john@penguin.example.com
john@penguin.example.com'spassword:
```

从安全命令行提示符中运行 X 程序，SSH 客户端与服务器创建一个新的安

全通道，X 程序数据通过该通道被透明地输送至客户机。

X11 转发程序可能非常有效。例如，X11 转发程序可用于创建打印机配置实用工具的一个新的安全、交互式会话。做法如下，请使用 ssh 与服务器连接，输入：

```
system-config-printer &
```

将出现打印机配置工具，使远程用户可以在远程系统上安全地配置打印。

13.4.2 端口转发

SSH 可通过端口转发保护其他不安全的 TCP/IP 协议。在使用此方法时，SSH 服务器将成为 SSH 客户端的加密导管。

端口传输的做法是将客户端上的本地端口映射到服务器上的远程端口。SSH 可将服务器上的任何端口映射到客户端上的任意端口。此方法无需端口数目匹配。



注意：预留端口数目的使用。

若要设置端口转发来监听 1024 以下的端口，需具备 root 用户访问权限。

要创建监听本地主机上连接的 TCP/IP 端口转发通道，请使用如下格式的指令：

```
ssh -L local-port:remote-hostname:remote-port username@hostname
```

例如，要通过加密连接使用 POP3 在服务器 mail.example.com 上查看邮件，请用如下指令：

```
ssh -L 1100:mail.example.com:110 mail.example.com
```

一旦在客户计算机及邮件服务器之间建立了端口转发通道，指示一个 POP3 邮件客户在本地主机上使用端口 1100 查看新邮件。客户系统上任何发至端口 1100 的请求将被安全地传输至 mail.example.com 服务器上。

若 mail.example.com 未运行 SSH 服务器，但同一网络中的另一台机器运行 SSH 服务器，仍然可使用 SSH 保护该连接。但要使用略微不同的指令：

```
ssh -L 1100:mail.example.com:110 other.example.com
```

在本例中，通过端口 22 与 SSH 服务器的 SSH 连接，来自客户计算机上端口 1100 的 POP3 请求被转发至 SSH 服务器 other.example.com。之后，other.example.com 连接至 mail.example.com 上的端口 110 来查看新邮件。注意，只有客户系统与 other.example.com SSH 服务器之间的连接是安全的，才可使用此方法。

端口转发也可通过网络防火墙来安全地接收信息。若将防火墙设置为允许 SSH 通过其标准端口传输信息（即端口 22）但阻止访问其他端口，则通过重定向已建立的 SSH 连接中的通信，仍然可能使用被阻断的端口建立两个主机间得连接。

连接仅与客户系统一样安全。

以此方式使用端口转发来转发连接，可以使客户端系统上的任意用户能连接至该服务。若客户端受到影响，则攻击者也有权访问被转发的服务。

担心端口转发的系统管理员可以禁用服务器上的此功能，只需通过指定 /etc/ssh/sshd_config 中的 AllowTcpForwarding 行的 No 参数并重启 sshd 服务。

13.5 额外信息资源

Open SSH 和 Open SSL 项目在不断发展，相关的最新信息可在他们的网站上找到。Open SSH 和 Open SSL 工具的帮助页面也是详细信息的丰富来源。

13.5.1 已安装文件

1) man ssh

ssh 的手册页面包含其使用方法的完整文件资料。

2) man scp

scp 的手册页面含有其使用方法的完整文件资料。

3) man sftp

sftp 的手册页面含有其使用方法的完整文件资料。

4) man sshd

sshd 的手册页面含有其使用方法的完整文件资料。

5) man ssh-keygen

ssh-keygen 的手册页面含有其使用方法的完整文件资料。

6) `man ssh_config`

手册页面包含可用的 SSH 客户配置选项的完整描述。

7) `man sshd_config`

手册页面包含可用的 SSH 后台程序配置选项的完整描述。

13.5.2 有用的网站

1) <http://www.openssh.com/>

Open SSH 首页有更多文件资料、常见问题、邮件列表链接、错误报告及其他有用信息资源。

2) <http://www.openssl.org/>

Open SSL 首页有更多文件资料、常见问题、邮件列表链接及其他有用信息资源。

3) <http://www.freesshd.com/>

另一种 SSH 服务器的部署。

14 DHCP 配置

DHCP (Dynamic Host Configuration Protocol, 动态主机配置协议) 是自动将 TCP/IP 信息分发至客户计算机的网络协议。各 DHCP 客户端连接到中央 DHCP 服务器, 中央 DHCP 服务器返回客户端的网络配置 (含 IP 地址、网关及 DNS 服务器)。

14.1 为什么要用 DHCP?

DHCP 对客户端网络接口的自动配置非常有用。在配置客户端系统时, 管理员选择 DHCP 即可, 而不用说明 IP 地址、网络掩码、网关或 DNS 服务器。客户可从 DHCP 服务器重新获取这些信息。若管理员想更改大量系统的 IP 地址, DHCP 也是非常有用的。他只需编辑服务器上的一个 DHCP 配置, 更改新 IP 地址集设置, 而不用重新设置所有的系统。若一个组织的 DNS 服务器发生变更, 该变更只发生在 DHCP 服务器上, 而非 DHCP 客户端。当管理员重启网络或重启客户端, 那些变更都会生效。

若一个组织将功能性服务器正确地连接到网络, 手提或其他移动计算机用户都可在办公室间移动这些设备。

14.2 DHCP 服务器的配置

dhcp 程序包带有 ISC DHCP 服务器。首先, 作为超级用户安装程序包。

```
yum install dhcp
```

安装 dhcp 程序包将新建一个空配置文件/etc/dhcp/dhcpd.conf:

```
cat /etc/dhcp/dhcpd.conf

DHCP Server Configuration file.

see /usr/share/doc/dhcp*/dhcpd.conf.sample
```

示例中的配置文件可在/usr/share/doc/dhcp-<version>/dhcpd.conf.sample 中找到。你应使用此文件来帮助你配置/etc/dhcp/dhcpd.conf, 做法在下文中详细解释。

DHCP 也利用文件/var/lib/dhcpd/dhcpd.leases 来存储客户租赁数据库。更多信息 14.2.2 租赁数据库。

14.2.1 配置文件

配置 DHCP 服务器的第一步是新建存储客户网络信息的配置文件。利用该文件来声明客户端系统的选项及全局选项。

配置文件可包含使格式化更容易的额外标签或空行。关键词不区分大小写，以(#)开头的行被当做解释。

配置文件中有两种形式的声明：

1) 参数

陈述如何执行一项任务，是否执行某项任务或将什么网络配置选项发送给客户。

2) 声明

描述网络拓补结构，描述客户端，为客户端提供地址、或将一组参数应用到一组声明上。

以关键字选项开始的参数被称为选项。这些选项控制 dhcp 选项，而参数配置那些非选择性或控制 DHCP 服务器行为的数值。

在({ })部分前声明的参数（含选项）被当做全局参数。全局参数适用于以下所有部分。



注意：重启 DHCP 守护程序，使变更生效

若配置文件发生变更，直到以 `service dhcpd -restart` 命令重启 DHCP 守护程序时，该变更方可生效。



提示：Omshell 指令的使用

使用 `omshell` 命令提供了一种交互式的方法来实现连接到、查询及变更为 DHCP 服务器的配置，无需每次更改 DHCP 配置文件并重启服务。使用 `omshell`，所有变更都可在服务器运行期间完成。更多有关 `omshell` 的信息，参见 `omshell` 的手册帮助页面。

在示例 14.1.中，“子网络声明”，路由器、子网络掩码、域名查询、域名服务器及时间偏置选项适用于任何以下的主机声明。

此外，也可声明子网络，必须包含网络中每个子网的子网络声明。否则，Dhcp 服务器将无法启动。

在本例中，声明了子网络中每个 DHCP 客户端的全局选项及范围。所有客户端都被分配了此范围内的 IP 地址。

例 14.1 子网络声明

```
subnet 192.168.1.0 netmask 255.255.255.0 {  
    option routers 192.168.1.254;  
    option subnet-mask 255.255.255.0;  
    option domain-search "example.com";  
    option domain-name-servers 192.168.1.1;  
    option time-offset-18000; # Eastern Standard Time  
range 192.168.1.10 192.168.1.100;  
}
```

若要配置一个 DHCP 服务器，此服务器将一个动态 IP 地址租给子网络内的一个系统，请将“例 14.1 范围参数”修改为您期望的值。它向客户声明默认租赁时间、最长租赁时间、网络配置值。本例向客户系统分配 192.168.1.10 与 192.168.1.100 之间的 IP 地址。

例 14.2 范围参数

```
default-lease-time 600;  
max-lease-time 7200;  
option subnet-mask 255.255.255.0;  
option broadcast-address 192.168.1.255;  
option routers 192.168.1.254;  
option domain-name-servers 192.168.1.1,192.168.1.2;  
option domain-search "example.com";  
subnet 192.168.1.0 netmask 255.255.255.0 {  
range 192.168.1.10 192.168.1.100;  
}
```

若要基于网络接口卡的 MAC 地址给客户端分配 IP 地址，请使用主机声明内 hardware ethernet 参数。如例 12.3“利用 Dhcp 的静态 IP 地址”中所述，host apex 声明指定 MAC 地址为 00:A0:78:8E:9E:AA 网络接口卡总是接收 IP 地址 192.168.1.4 的数据。



注意：可选的参数-主机名-也可被用于分配主机名给客户端。

例 14.3 使用 DHCP 的静态 IP 地址

```
host apex{  
option host-name "apex.example.com";
```

```
hardware ethernet 00:A0:78:8E:9E:AA;
fixed-address 192.168.1.4;
}
```

如例 14.4.“共享网络声明”中所述，所有共享同一物理网络的子网络都应在共享网络声明内进行声明。在共享网络内但处于封闭子网络声明外的参数，被当做全局参数。共享网络名应为网络的描述性名称，如用“测试实验室”来描述所有测试实验室环境中的子网络。

例 14.4 共享网络声明

```
shared-networkname {
option  domain-search      "test.redhat.com";
option  domain-name-servers ns1.redhat.com, ns2.redhat.com;
option  routers            192.168.0.254;
more parameters for EXAMPLE shared-network
subnet 192.168.1.0 netmask 255.255.252.0 {
parameters for subnet
range 192.168.1.1 192.168.1.254;
}
subnet 192.168.2.0 netmask 255.255.252.0 {
parameters for subnet
range 192.168.2.1 192.168.2.254;
}
}
```

如例 14.5.“分组声明”中所述，分组声明用于将全局参数应用至一组声明中。例如，共享网络、子网络及主机都可被分组。

例 14.5 分组声明

```
group {
option  routers    192.168.1.254;
option  subnet-mask 255.255.255.0;
option  domain-search "example.com";
option  domain-name-servers 192.168.1.1;
option  time-offset -18000;      # Eastern Standard Time
host apex{
option host-name "apex.example.com";
hardware ethernet 00:A0:78:8E:9E:AA;
```

```
fixed-address 192.168.1.4;
    }
hostraleigh {
option host-name "raleigh.example.com";
hardware ethernet 00:A1:DD:74:C3:F2;
fixed-address 192.168.1.6;
    }
}
```

注意：使用示例配置文件。

示例中的配置文件可用作配置文件的起点，可将自定义配置选项添加到此文件中。若要将其复制至合适位置，请用如下命令：

```
cp /usr/share/doc/dhcp-<version-number>/dhcpd.conf.sample /etc/dhcp/dhcpd.conf
```

其中<version-number>是 DHCP 的版本号。

选项声明的完整列表及各自的功能，参见 `dhcp-options` 手册帮助页面。

14.2.2 租赁数据库

DHCP 服务器上，文件 `/var/lib/dhcpd/dhcpd.leases` 存档 DHCP 客户租赁数据库。不要修改此文件。因为每个最近被分配 IP 地址的 Dhcp 租赁信息都自动存储在租赁数据库中。该信息包含租赁期限、IP 地址分配给谁、租赁的起始日期及用于重新获取租赁的网络接口卡的 MAC 地址。

租赁数据库中的所有时间为协调通用时间而非当地时间。

租赁数据库不时被重建以使其不致过大。具体过程为：首先，所有已知租赁被存储于临时租赁数据库中。`dhcpd.leases` 文件被重新命名为 `dhcpd.leases~`，临时租赁数据库被写入 `dhcpd.leases`。

在租赁数据库被重命名为备份文件后但在新文件被写入前，DHCP 守护程序可能被杀死，或系统可能崩溃。若发生此情况，`dhcpd.leases` 文件不再存在，但是需要重新启动服务。不要新建一个新租赁文件。若这样做，所有旧租赁将丢失而导致很多问题。正确的做法是将备份文件 `dhcpd.leases~` 重命名为 `dhcpd.leases`，然后启动守护程序。

14.2.3 启动和终止服务器



注意：初次启动 DHCP 服务器

当初次启动 DHCP 服务器，除非 `dhcpd.leases` 文件存在，否则将会启动失败。若该文档不存在，使用指令 `touch /var/lib/dhcpd/dhcpd.leases` 来创建该文档。

若同一服务器也将 BIND 作为 DNS 服务器运行，此步骤不是必要的，因为启动命名服务将自动查找 `dhcpd.leases` 文件。

要启动 DHCP 服务，使用指令 `/sbin/service dhcpd start.`。要终止 DHCP 服务器，使用指令 `/sbin/service dhcpd stop.`

默认设置是，DHCP 服务不在启动时运行。

若系统有不只一个网卡，但 DHCP 服务器仅可在一个接口上启动，将 DHCP 服务器设置为仅在该设备上启动。在 `/etc/sysconfig/dhcpd` 文件中，将该接口名称添加至 `DHCPDARGS` 表单中：

```
Command line optionshere
DHCPDARGS=eth0
```

这对带有两个网卡的防火墙计算机非常有用。一张网卡可设置为 DHCP 客户端来重新获取连接因特网的 IP 地址。另一网卡可作为防火墙后面的内部网络的 DHCP 服务器。仅说明与内部网络连接的网卡使系统更安全，因为用户不可通过因特网与守护程序连接。

其他可在 `/etc/sysconfig/dhcpd` 中指定的指令行选项包含：

1) `-p <portnum>`

说明 `dhcpd` 应监听的 UDP 端口编号。默认编号为端口 67。DHCP 服务器向处于端口编号比规定的 UDP 端口编号更大的 DHCP 客户端传送响应。例如，使用默认端口 67，服务器在端口 67 监听端口 68 号客户端的请求和响应。若此处规定端口并使用 DHCP 中继代理，则也必须指定 DHCP 中继代理应监听的同一端口。详细信息参见 14.2.4 DHCP 中继代理。

2) `-f`

将守护程序作为前台进程运行。多用于调试。

3) `-d`

将 DHCP 服务器守护程序登录至标准错误信息描述器。多用于调试。若未

做说明，该记录写入/var/log/messages。

4) -cf <filename>

说明配置文件的存储位置。默认位置为/etc/dhcp/dhcpd.conf。

5) -lf<filename>

说明租赁数据库文件的位置。若租赁数据库文件已存在，每次 DHCP 服务器启动时使用同一文件非常重要。强烈建议此选项仅用于非生产机器的调试。默认位置为/var/lib/dhcpd/dhcpd.leases。

6) -q

启动守护程序时不打印整个版权信息。

14.2.4 DHCP 中继代理

DHCP 中继代理(dhcrelay)允许 DHCP 中继，以及从没有 DHCP 服务器的子网发送到其他子网的一个或多个 DHCP 服务器的 BOOTP 请求。

当一个 DHCP 客户端发出信息请求，DHCP 中继代理将请求发送至 DHCP 中继代理启动时指定的 DHCP 服务器列表。当 DHCP 服务器给出响应时，该响应在发出原请求的网络中被广播或单播。

DHCP 中继代理监听所有接口上的 DHCP 请求，除非通过 INTERFACES 指令在/etc/sysconfig/dhcrelay 中指定接口。

要启动 DHCP 中继代理，请使用 service dhcrelay start。

14.3 配置 DHCP 客户端

若要手动配置 DHCP 客户端，请修改/etc/sysconfig/network 文件来启动网络连接和/etc/sysconfig/network-scripts 目录中各个网络设备的配置文件。在该目录中，各设备应带名为 ifcfg-eth0 的配置文件，其中 eth0 为网络设备的名称。

/etc/sysconfig/network-scripts/ifcfg-eth0 文件应包含以下：

```
DEVICE="eth0"
BOOTPROTO=dhcp
ONBOOT=yes
```

每个设备都必须有一个配置文件，该配置文件配置为使用 DHCP。

网络脚本的其他选项包含：

DHCP_HOSTNAME——只有在 DHCP 服务器要求客户端在接受 IP 地址前指定主机名称时使用此选项。

PEERDNS=<answer>, <answer>为以下之一:

- 1) Yes——用服务器上的信息修正/etc/resolv.conf。若使用 DHCP, yes 为默认回答。
- 2) No——不修正/etc/resolv.conf。



注意: 高级配置:

客户端 DHCP 选项的高级配置, 例如协议时间、租赁要求及请求、动态 DNS 支持、别名、以及重载、预先计划、或添加到客户端配置的各种各样的数值, 参见 dhclient 与 shclient.conf 的帮助页面。

14.4 多地址 DHCP 服务器的配置

多地址 DHCP 服务器为多个网络服务, 即, 多个子网络。本节中的案例详述如何配置 DHCP 服务器以服务多个网络, 选择监听哪个网络接口、及如何为移动网络定义网络设置。

在做任何更改前, 将现有的/etc/sysconfig/dhcpd 与/etc/dhcp/dhcpd.conf 进行备份。

除非另有说明, DHCP 守护程序监听所有的网络接口。使用/etc/sysconfig/dhcpd 文件指定 DHCP 守护程序需要监听的网络接口。下文中/etc/sysconfig/dhcpd 的范例说明 DHCP 守护程序监听 eth0 与 eth1 接口:

```
DHCPDARGS="eth0 eth1";
```

若系统有三个网络接口卡——eth0、eth1 和 eth2, 只希望 DHCP 守护程序监听 eth0, 那么仅在/etc/sysconfig/dhcpd 中指定 eth0 即可。

```
DHCPDARGS="eth0";
```

以下是基本/etc/dhcp/dhcpd.conf 文件, 该服务器有两个网络接口, 10.0.0.0/24 网络中的 eth0 及 172.16.0.0/24 网络中的 eth1。多个子网络声明允许为多个网络定义不同设置:

```
default-lease-time 600;
max-lease-time 7200;
subnet 10.0.0.0 netmask 255.255.255.0{
```

```
option subnet-mask255.255.255.0;
option routers10.0.0.1;
range10.0.0.510.0.0.15;
}
subnet 172.16.0.0 netmask 255.255.255.0{
option subnet-mask255.255.255.0;
option routers 172.16.0.1;
range172.16.0.5172.16.0.15;
}
```

Subnet 子网络 10.0.0.0 netmask 网络掩码 255.255.255.0;

DHCP 服务器服务的每个网络都应具备子网络声明。多个子网络要求具备多个子网络声明。若 DHCP 服务器无子网络声明范围内的网络接口，那么 DHCP 服务器不为该网络服务。

若仅有一个子网络声明，且在该子网络范围内无网络接口，那么 DHCP 守护程序无法启动，且将如下错误被记录到/var/log/messages 中：

```
dhcpcd:No subnetdeclaration foreth0(0.0.0.0).
dhcpcd:  **    Ignoring requests on eth0. Ifthisisnot what
dhcpcd:  you   want,pleasewriteasubnetdeclaration
dhcpcd:  in yourdhcpcd.conf file for the network segment
dhcpcd:  to which interface eth1is attached. **
dhcpcd:
dhcpcd:
dhcpcd:  Not   configured to listen on any interfaces!
```

option subnet-mask 选项子网络掩码 255.255.255.0;

选项子网络掩码选项定义子网络掩码，并在子网络声明中重载网络掩码值。在简单情况下，子网络与网络掩码值相同。

option routers 选项路由器 10.0.0.1;

选项路由器选项定义子网络的默认网关。系统要在不同的子网络上连接内部网络及外部网络时，应具备该选项。

Range 范围 10.0.0.5 10.0.0.15;

范围选项规定可用 IP 地址池。系统被分配给规定 IP 地址范围中的一个 IP 地址。

更多信息参见 `dhcpd.conf(5)` 帮助页面。

 警告：不要用别名接口

DHCP 不支持别名接口。若别名接口是 `/etc/dhcp/dhcpd.conf` 指定的子网络中的唯一接口，DHCP 守护程序将无法启动。

14.4.1 主机配置

在做任何变更前，都请备份 `/etc/sysconfig/dhcpd` 与 `/etc/dhcp/dhcpd.conf` 文件。

为多个网络配置单一系统

以下 `/etc/dhcp/dhcpd.conf` 范例创建两个子网络并为同一系统配置一个 IP 地址，这取决于其连接哪个网络。

```
default-lease-time600;
max-lease-time7200;
subnet10.0.0.0 netmask255.255.255.0{
option subnet-mask255.255.255.0;
optionrouters10.0.0.1;
range10.0.0.510.0.0.15;
}
subnet 172.16.0.0 netmask 255.255.255.0{
option subnet-mask255.255.255.0;
optionrouters 172.16.0.1;
range172.16.0.5172.16.0.15;
}
host example0 {
hardwareethernet 00:1A:6B:6A:2E:0B;
fixed-address10.0.0.20;
}
host example1 {
hardwareethernet 00:1A:6B:6A:2E:0B;
fixed-address172.16.0.20;
}
host example0
```

主机声明定义单一系统的具体参数，例如 IP 地址。要为多个主机配置具体参数，请使用多个主机声明。多数 DHCP 客户忽视主机声明中的名称，同样的，此名称可为任意名称，只要它对主机声明来讲是唯一的。为多个网络配置同样的

系统，请为每个主机声明使用不同的名称，否则 DHCP 守护程序将无法启动。系统是通过 hardware ethernet 选项被鉴别出来的，而通过非主机声明的名称。

```
hardware ethernet 00:1A:6B:6A:2E:0B;
```

hardware ethernet 选项识别系统。要找到此地址，请运行 ip link 命令。

```
fixed-address 10.0.0.20;
```

固定地址选项为 hardware ethernet 选项指定的系统分配一个有效 IP 地址。该地址需在由范围选项规定的 IP 地址池之外。

若范围声明不是以分号结尾，则 DHCP 守护程序无法启动，如下错误将记录至 /var/log/messages:

```
/etc/dhcp/dhcpd.conf line 20:semicolon expected.
dhcpd: }
dhcpd: ^
dhcpd:/etc/dhcp/dhcpd.conf line 38:unexpected end of file
dhcpd:
dhcpd: ^
dhcpd: Configuration file errors encountered--exiting
```

配置有多个网络接口的系统。

如下主机声明配置带有多个网络接口的系统，以使各接口接收同样的 IP 地址。若两个网络接口同时连接同一网络，该配置将失效。

```
host interface0 {
    hardware ethernet 00:1a:6b:6a:2e:0b;
    fixed-address 10.0.0.18;
}
host interface1 {
    hardware ethernet 00:1A:6B:6A:27:3A;
    fixed-address 10.0.0.18;
}
```

本范例中，接口 0 是第一个网络接口，而接口 1 是第二个接口。不同 hardware ethernet 识别不同接口。

若这样的系统与另一网络连接，需再增加主机声明，请记住：

为主机连接到的网络提供一个有效的固定 IP 地址。

为主机声明取一个特殊的名称。

若主机声明的名称不唯一，DHCP 守护程序将无法启动，如下错误将记录至 /var/log/messages。

```
dhcpcd:/etc/dhcp/dhcpd.conf:line 31: host interface0: already exists
dhcpcd: }
dhcpcd: ^
dhcpcd: Configuration file errors encountered--exiting
```

此错误是由在 /etc/dhcp/dhcpd.conf 中定义多个主机接口 0 声明引起的。

14.5 DHCP for IPv6 (DHCPv6)

ISC DHCP 包含对 IPv6 的支持 (DHCPv6)，因为从 4x 版本开始就包含了 DHCPv6 服务器、客户端及中继代理功能。服务器、客户端及中继代理同时支持 IPv4 与 IPv6。然而，客户端与服务器仅可一次处理一份协议，若要做到双重支持，需分别为 IPv4 与 IPv6 启动它们。

DHCPv6 服务器配置可在 /etc/dhcp/dhcpd6.conf 文件中找到。

示例服务器配置文件可在 /usr/share/doc/dhcp-<version>/dhcpd6.conf.sample 中找到。

要启动 DHCPv6 服务，使用指令 /sbin/service dhcpd6 start。

一个简单的 DHCPv6 服务器配置文件如下：

```
subnet6 2001:db8:0:1::/64 {
    range6 2001:db8:0:1::129 2001:db8:0:1::254;
    option dhcp6.name-servers fec0:0:0:1::1;
    option dhcp6.domain-search "domain.example";
}
```

14.6 更多信息资源

更多信息参见 The DHCP Handbook: Ralph Droms and Ted Lemon, 2003 或如下信息资源。

14.6.1 已安装的文件

- 1) dhcpd 帮助页面——描述 DHCP 守护程序如何工作。
- 2) dhcpd.conf 帮助页面——解释如何设置 DHCP 配置文件，含例子。

- 3) `dhcpd.leases` 帮助页面——描述一个持久的租赁数据库。
- 4) `dhcp`——选项帮助页面——解释说明 `dhcpd.conf` 中的 DHCP 选项的句法。含例子。
- 5) `dhcrelay` 帮助页面——解释 DHCP 中继代理及其配置选项。
- 6) `/usr/share/doc/dhcp-<version>/`——含范例文件，`README` 文件及 DHCP 服务的当前版本的发行说明。

15 MYSQL 配置和使用

MySQL 是一个精巧的 SQL 数据库管理系统,虽然它不是开放源代码的产品,但在某些情况下您可以自由使用。由于它的强大功能、灵活性、丰富的应用编程接口 (API) 以及精巧的系统结构,受到了广大自由软件爱好者甚至是商业软件用户的青睐,特别是与 Apache 和 PHP/PERL 结合,为建立基于数据库的动态网站提供了强大动力。

MySQL 由瑞典的 T.c.X 公司负责开发和维护,MySQL 的用户手册很单纯,只有一个集中的<MySQL Reference Manual>,但其内容覆盖了 MySQL 的所有信息,因此该手册是了解和掌握 MySQL 的绝佳文献。

MySQL 官方站点: <http://www.mysql.com>。

15.1 MYSQL 的主要特性

下面介绍了 MySQL 数据库软件的一些重要特性。

- 1) 内部构件和可移植性。
- 2) 使用 C 和 C++编写。
- 3) 用众多不同的编译器进行了测试。
- 4) 能够工作在众多不同的平台上。
- 5) 使用 GNU Automake、Autoconf 和 Libtool 进行移植。
- 6) 提供了用于 C、C++、Eiffel、Java、Perl、PHP、Python、Ruby 和 Tcl 的 API。
- 7) 采用核心线程的完全多线程如果有多个 CPU,它能方便地使用这些 CPU。
- 8) 提供了事务性和非事务性存储引擎。
- 9) 使用了极快的“B 树”磁盘表 (MyISAM) 和索引压缩。
- 10) 添加另一个存储引擎相对简单。如果打算为内部数据库添加一个 SQL 接口,该特性十分有用。
- 11) 极快的基于线程的内存分配系统。
- 12) 通过使用优化的“单扫描多连接”,能实现极快的连接。
- 13) 存储器中的哈希表用作临时表。

- 14) SQL 函数是使用高度优化的类库实现的，运行很快。通常，在完成查询初始化后，不存在存储器分配。
- 15) 采用 Purify（商业内存溢出检测器）以及 GPL 工具 Valgrind (<http://developer.kde.org/~sewardj/>) 测试了 MySQL 代码。
- 16) 服务器可作为单独程序运行在客户端/服务器联网环境下。它也可作为库提供，可嵌入（链接）到独立的应用程序中。这类应用程序可单独使用，也能在网络环境下使用。
- 17) 列类型
 - a) 众多列类型：带符号/无符号整数，1、2、3、4、8 字节长，FLOAT，DOUBLE，CHAR，VARCHAR，TEXT，BLOB，DATE，TIME，DATETIME，TIMESTAMP，YEAR，SET，ENUM，以及 OpenGIS 空间类型。
 - b) 定长和可变长度记录。
- 18) 语句和函数
 - a) 在 SELECT 和查询的 WHERE 子句中，提供完整的操作符和函数支持。例如：

```
mysql> SELECT CONCAT(first_name, ' ', last_name)
-> FROM citizen
-> WHERE income/dependents > 10000 AND age > 30;
```

- b) 对 SQL GROUP BY 和 ORDER BY 子句的全面支持。支持聚合函数 (COUNT(), COUNT(DISTINCT ...), AVG(), STD(), SUM(), MAX(), MIN()和 GROUP_CONCAT())。
 - c) 支持 LEFT OUTER JOIN 和 RIGHT OUTER JOIN，采用标准的 SQL 和 ODBC 语法。
 - d) 按照标准 SQL 的要求，支持表别名和列别名。
 - e) DELETE、INSERT、REPLACE 和 UPDATE 返回更改（影响）的行数。连接到服务器时，可通过设置标志返回匹配的行数。
 - f) MySQL 的 SHOW 命令可用于检索关于数据库、数据库引擎、表和索引的信息。EXPLAIN 命令可用于确定优化器处理查询的方式。
 - g) 函数名与表名或列名不冲突。例如，ABS 是有效的列名。唯一的限

制在于，调用函数时，函数名和随后的符号“(”之间不得有空格。

- h) 可以将不同数据库的表混合在相同的查询中(就像 MySQL 3.22 中那样)。

19) 安全

- a) 十分灵活和安全的权限和密码系统，允许基于主机的验证。连接到服务器时，所有的密码传输均采用加密形式，从而保证了密码安全。

20) 可伸缩性和限制

- a) 处理大型数据库: 我们使用了 MySQL 服务器和含 5 千万条记录的数据库。我们还听说，有些用户将 MySQL 用于含 60000 个表和约 50 亿行的数据库。
- b) 每个表可支持高达 64 条索引(在 MySQL 4.1.2 之前为 32 条)。每条索引可由 1~16 个列或列元素组成。最大索引宽度为 1000 字节(在 MySQL 4.1.2 之前为 500)。索引可使用具备 CHAR、VARCHAR、BLOB 或 TEXT 列类型的列前缀。

21) 连接性

- c) 在任何平台上，客户端可使用 TCP/IP 协议连接到 MySQL 服务器。在 Windows 系统的 NT 系列中(NT、2000、XP 或 2003)，客户端可使用命名管道进行连接。在 Unix 系统中，客户端可使用 Unix 域套接字文件建立连接。
- d) 在 MySQL 4.1 和更高的版本中，如果是以“--shared-memory”选项开始，Windows 服务器还支持共享内存连接。客户端可使用“--protocol=memory”选项，通过共享内存建立连接。
- e) Connector/ODBC (MyODBC)接口为使用 ODBC(开放式数据库连接性)连接的客户端程序提供了 MySQL 支持。例如，可以使用 MS Access 连接到您的 MySQL 服务器。客户端可运行在 Windows 或 Unix 平台上。提供了 MyODBC 源。支持所有的 ODBC 2.5 函数，以及众多其他函数。
- f) Connector/J 接口为使用 JDBC 连接的 Java 客户端程序提供了 MySQL 支持。客户端可运行在 Windows 或 Unix 平台上。提供了 Connector/J

源码。

22) 本地化

- a) 服务器可使用多种语言向客户端提供错误消息。
- b) 对数种不同字符集的全面支持，包括 latin1 (cp1252)、german、big5、ujis 等。例如，在表名和列名中允许使用斯堪的纳维亚字符 “ä”、“å” 和 “ö”。从 MySQL 4.1 开始，提供了 Unicode 支持。
- c) 所有数据均以所选的字符集保存。正常字符串列的比较不区分大小写。
- d) 分类是根据所选的字符集（默认情况下，使用瑞典校对）进行的。启动 MySQL 服务器时，可更改该项设置。要想查看高级分类的示例，请参见 Czech 分类代码。MySQL 服务器支持众多不同的字符集，这类字符集可在编译时和运行时指定。

23) 客户端和工具

- a) MySQL 服务器提供了对 SQL 语句的内部支持，可用于检查、优化和修复表。通过 mysqlcheck 客户端，可在命令行上使用这类语句。MySQL 还包括 myisamchk，这是一种很快的命令行实用工具，可用于在 MyISAM 表上执行这类操作。
- b) 对于所有 MySQL 程序，均能通过--help 或--?选项调用，以获取联机帮助信息。

15.2 数据库管理

15.2.1 MYSQL 服务器和服务启动脚本

MySQL 服务器，即 mysqld，是在 MySQL 安装中负责大部分工作的主程序。服务器随附了几个相关脚本，当您安装 MySQL 时它们可以执行设置操作，或者是帮助您启动和停止服务器的帮助程序：

```
shell> service mysqld start  
shell> service mysqld stop
```

15.2.2 服务器端脚本和实用工具概述

MySQL 程序采用各种不同的选项。但每个 MySQL 程序提供一个--help 选项，您可以用来查阅程序选项相关说明。例如，您可以试试 mysqld --help。

您可以在命令行中或在选项文件中指定选项来替换所有标准程序中的默认选项。

下面简单描述了 MySQL 服务器和服务器相关程序：

1) `mysqld`

SQL 后台程序(即 MySQL 服务器)。要想使用客户端程序，该程序必须运行，因为客户端通过连接服务器来访问数据库。

2) `mysqld-max`

包括更多特性的一个服务器版本。

3) `mysqld_safe`

服务器启动脚本。如果 `mysqld-max` 存在，`mysqld_safe` 试图启动它，否则启动 `mysqld`。

4) `mysql.server`

服务器启动脚本。该脚本用于使用包含为特定级别的运行启动服务的脚本的运行目录的系统。它调用 `mysqld_safe` 来启动 MySQL 服务器。

5) `mysqld_multi`

服务器启动脚本，可以启动或停止系统上安装的多个服务器。

6) `mysql_install_db`

该脚本用默认权限创建 MySQL 授权表。通常只是在系统上首次安装 MySQL 时执行一次。

7) `mysql_fix_privilege_tables`

在升级安装后，如果新版本 MySQL 中的授权表有更改，则使用该脚本来更改授权表。

服务器主机上还运行其它几个程序：

8) `myisamchk`

用来描述、检查、优化和维护 MyISAM 表的实用工具。

9) `make_binary_distribution`

该程序可以生成编译过的 MySQL 的二进制版本。可以通过 FTP 上传到 `ftp.mysql.com` 的 `/pub/mysql/upload/`，供其它 MySQL 用户使用。

10) `mysqlbug`

MySQL 缺陷报告脚本。它可以用来向 MySQL 邮件系统发送缺陷报告。(您也可以访问 <http://bugs.mysql.com/> 在线创建缺陷报告文件。

15.2.3 MYSQLD-MAX 扩展 MYSQL 服务器

MySQL-Max 服务器是 MySQL 服务器的一个版本，包含了更多的特性。该分发版的使用取决于您的平台：

对于 Linux，如果您使用 RPM 分发版安装 MySQL，首先使用常规 MySQL-server RPM 来安装标准 mysqld 服务器。然后使用 MySQL-Max RPM 来安装 mysqld-max 服务器。MySQL-Max RPM 假定您已经安装了常规服务器 RPM。

所有其它 MySQL-Max 分发版包含一个 mysqld 服务器，但具有更多的特性。

MySQL AB 使用下面的 configure 选项构建 MySQL-Max 服务器：

1) `--with-server-suffix=-max`

该选项为 mysqld 版本字符串添加一个 -max 后缀。

2) `--with-innodb`

该选项启用 InnoDB 存储引擎支持。MySQL-Max 服务器包括 InnoDB 支持。在 MySQL 4.0 及以上版本中，默认 InnoDB 包括在所有二进制分发版中，因此您不需要用 MySQL-Max 服务器只是用来获取 InnoDB 支持。

3) `--with-bdb`

该选项启用 Berkeley DB (BDB) 存储引擎支持。

4) `--with-blackhole-storage-engine`

该选项启用 BLACKHOLE 存储引擎支持。

5) `USE_SYMDIR`

启用该定义来为 Windows 打开数据库符号链接支持。符号链接支持适用于所有 Windows 服务器，因此 Max 服务器不需要支持该特性。

6) `--with-ndbcluster`

该选项启用 NDB Cluster 存储引擎支持。目前(5.1.2-alpha)只有 Linux、Solaris 和 Mac OS X 支持 Cluster。已有一些用户报告在 BSD 操作系统上成功使用了从源码构建的 MySQL Cluster，但目前还没有得到官方支持。

MySQL-Max 二进制分发版对于想要安装预编译程序的用户很方便。如果您使用源码分发版构建 MySQL，您可以通过在配置时启用 MySQL-Max 二进制分

发版构建所用的相同的特性来构建您自己的 Max-like 服务器。

MySQL-Max 服务器包括 BerkeleyDB (BDB)存储引擎，但并非所有平台支持 BDB。

Solaris、Mac OS X 和 Linux(在大多数平台上)的 MySQL-Max 服务器包括 NDB CLUSTER 存储引擎支持。请注意必须用 ndbcluster 选项启动服务器，以便使服务器做为 MySQL Cluster 的一部分来运行。表格 15-1 显示了 MySQL-Max 二进制在哪个平台上包括 BDB 和/或 NDB CLUSTER 支持：

表格 15-1 系统支持

系统	BDB 支持	NDB 支持
AIX 4.3	N	N
HP-UX 11.0	N	N
Linux-Alpha	N	Y
Linux-IA-64	N	N
Linux-Intel	Y	Y
Mac OS X	N	N
NetWare	N	N
SCO OSR5	Y	N
Solaris-SPARC	Y	Y
Solaris-Intel	N	Y
UnixWare	Y	N
Windows NT/2000/XP	Y	N

要想找出您的服务器支持哪个存储引擎，执行下面的语句：

```
mysql> SHOW ENGINES;
+-----+-----+-----+
| Engine | Support | Comment |
+-----+-----+-----+
| MyISAM | DEFAULT | Default engine as of MySQL 3.23 with great performance |
| MEMORY | YES | Hash based, stored in memory, useful for temporary tables |
```

```
| HEAP | YES | Alias for MEMORY |
| MERGE | YES | Collection of identical MyISAM tables |
| MRG_MYISAM | YES | Alias for MERGE |
| ISAM | NO | Obsolete storage engine, now replaced by MyISAM |
| MRG_ISAM | NO | Obsolete storage engine, now replaced by MERGE |
| InnoDB | YES | Supports transactions, row-level locking, and foreign keys |
| INNODB | YES | Alias for INNODB |
| BDB | YES | Supports transactions and page-level locking |
| BERKELEYDB | YES | Alias for BDB |
| NDBCLUSTER | NO | Clustered, fault-tolerant, memory-based tables |
| NDB | NO | Alias for NDBCLUSTER |
| EXAMPLE | NO | Example storage engine |
| ARCHIVE | YES | Archive storage engine |
| CSV | NO | CSV storage engine |
| FEDERATED | YES | Federated MySQL storage engine |
| BLACKHOLE | YES | /dev/null storage engine (anything you write to it disappears) |
+-----+-----+-----+
18 rows in set (0.00 sec)
```

您还可以使用下面的语句代替 **SHOW ENGINES**，并检查您感兴趣的存储引擎的变量值：

```
mysql> SHOW VARIABLES LIKE 'have%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_archive | YES |
| have_bdb | NO |
| have_blackhole_engine | YES |
| have_compress | YES |
| have_crypt | YES |
| have_csv | YES |
| have_example_engine | NO |
| have_federated_engine | NO |
| have_geometry | YES |
| have_innodb | YES |
| have_isam | NO |
```

```
| have_ndbcluster | DISABLED |
| have_openssl | NO |
| have_partition_engine | YES |
| have_query_cache | YES |
| have_raid | NO |
| have_rtree_keys | YES |
| have_symlink | YES |
+-----+-----+
18 rows in set (0.01 sec)
```

SHOW 命令的精确输出随使用的 MySQL 版本(和启用的特性)的不同而有变化。第 2 列的值表示各特性支持的服务器级别，如所示：

表格 15-2 各特性支持的服务器级别

值	含义
YES	支持该特性并已经激活。
NO	不支持该特性。
DISABLED	支持该特性但被禁用。

NO 值表示编译的服务器不支持该特性，因此在运行时不能激活。

出现 DISABLED 值是因为服务器启动时该特性被禁用，或没有给出启用它的所有选项。在后一种情况，host_名.err 错误日志文件应包含该选项被禁用的原因。

如果服务器支持 InnoDB 或 BDB 存储引擎，您还可以看见 DISABLED，但在运行启动时使用了--skip-innodb 或--skip-bdb 选项。对于 NDB CLUSTER 存储引擎，DISABLED 表示服务器支持 MySQL Cluster，但启动时未启用--ndb-cluster 选项。

所有 MySQL 服务器支持 MyISAM 表，因为 MyISAM 是默认存储引擎。

15.2.4 MYSQLD_SAFE: MYSQL 服务器启动脚本

在 Unix 和 NetWare 中推荐使用 mysqld_safe 来启动 mysqld 服务器。mysqld_safe 增加了一些安全特性，例如当出现错误时重启服务器并向错误日志文件写入运行时间信息。本节后面列出了 NetWare 的特定行为。

 注意：为了保持同旧版本 MySQL 的向后兼容性，MySQL 二进制分发版仍然包括

safe_mysql 作为 mysql_safe 的符号链接。但是，您不应再依赖它，因为在将来将删掉它。

默认情况下，mysql_safe 尝试启动可执行 mysql-max（如果存在），否则启动 mysql。该行为的含义是：

在 Linux 中，MySQL-Max RPM 依赖该 mysql_safe 的行为。RPM 安装可执行 mysql-max，使 mysql_safe 从该点起自动使用可执行命令。

如果您安装包括 mysql-max 服务器的 MySQL-Max 分发版，后面升级到非-Max 的 MySQL 版本，mysql_safe 仍然试图运行旧的 mysql-max 服务器。升级时，您应手动删除旧的 mysql-max 服务器以确保 mysql_safe 运行新的 mysql 服务器。

要想越过默认行为并显式指定您想要运行哪个服务器，为 mysql_safe 指定 --mysql 或 --mysql-version 选项。

mysql_safe 的许多选项与 mysql 是相同的。

所有在命令行中为 mysql_safe 指定的选项被传递给 mysql。如果您想要使用 mysql 不支持的 mysql_safe 的选项，不要在命令行中指定。相反，在选项文件的[mysql_safe]组内将它们列出来。

mysql_safe 从选项文件的[mysql]、[server]和[mysql_safe]部分读取所有选项。为了保证向后兼容性，它还读取[safe_mysql]部分，尽管在 MySQL 5.1 安装中您应将这部分重新命名为[mysql_safe]。

mysql_safe 支持下面的选项：

1) --help

显示帮助消息并退出。

2) --autoclose

(只在 NetWare 中)在 NetWare 中，mysql_safe 可以保持窗口。当您关掉 mysql_safe NLM 时，窗口不按默认设置消失。相反，它提示用户输入：

```
*<NLM has terminated; Press any key to close the screen>*
```

如果您想让 NetWare 自动关闭窗口，在 mysql_safe 中使用 --autoclose 选项。

3) --basedir=path

MySQL 安装目录的路径。

4) `-core-file-size=size`

mysqld 能够创建的内核文件的大小。选项值传递给 `ulimit -c`。

5) `--datadir=path`

数据目录的路径。

6) `--defaults-extra-file=path`

除了通用选项文件所读取的选项文件名。如果给出，必须首选该选项。

7) `--defaults-file=path`

读取的代替通用选项文件的选项文件名。如果给出，必须首选该选项。

8) `--ledir=path`

包含 `mysqld` 程序的目录的路径。使用该选项来显式表示服务器位置。

9) `--log-error=path`

将错误日志写入给定的文件。

10) `--mysqld=prog_name`

想要启动的服务器程序名(在 `ledir` 目录)。如果您使用 MySQL 二进制分发版但有二进制分发版之外的数据目录需要该选项。

11) `--mysqld-version =suffix`

该选项类似 `--mysqld` 选项，但您只指定服务器程序名的后缀。基本名假定为 `mysqld`。例如，如果您使用 `--mysqld-version =max`，`mysqld_safe` 启动 `ledir` 目录中的 `mysqld-max` 程序。如果 `--mysqld-version` 的参数为空，`mysqld_safe` 使用目录中的 `mysqld`。

12) `--nice=priority`

使用 `nice` 程序根据给定值来设置服务器的调度优先级。

13) `--no-defaults`

不要读任何选项文件。如果给出，必须首选该选项。

14) `--open-files-limit=count`

mysqld 能够打开的文件的数量。选项值传递给 `ulimit-n`。请注意您需要用 `root` 启动 `mysqld_safe` 来保证正确工作！

15) `--pid-file=path`

进程 ID 文件的路径。

16) `--port=port_num`

用来监听 TCP/IP 连接的端口号。端口号必须为 1024 或更大值,除非 MySQL 以 root 系统用户运行。

17) `--skip-character-set-client-handshake`

忽略客户端发送的字符集信息,使用服务器的默认字符集。(选择该选项,MySQL 的动作与 MySQL 4.0 相同)。

18) `--socket=path`

用于本地连接的 Unix 套接字文件。

19) `--timezone=zone`

为给定的选项值设置 TZ 时区环境变量。从操作系统文档查阅合法的时区规定格式。

20) `--user={user_name | user_id}`

以用户名 `user_name` 或数字用户 ID `user_id` 运行 `mysqld` 服务器。(本文中的用户指系统登录账户,而不是授权表中的 MySQL 用户)。

执行 `mysqld_safe` 时,必须先给出 `--defaults-file` 或 `--defaults-extra-option`,或不使用选项文件。例如,该命令将不使用选项文件:

```
mysqld_safe --port=port_num --defaults-file=file_name
```

相反,使用下面的命令:

```
mysqld_safe --defaults-file=file_name --port=port_num
```

一般情况 `mysqld_safe` 脚本可以启动从源码或二进制 MySQL 分发版安装的服务器,即使这些分发版将服务器安装到稍微不同的位置。`mysqld_safe` 期望下面的其中一个条件是真的:

可以根据调用 `mysqld_safe` 的目录找到服务器和数据库。在二进制分发版中,`mysqld_safe` 看上去在 `bin` 和 `data` 目录的工作目录下。对于源码分发版,为 `libexec` 和 `var` 目录。如果您从 MySQL 安装目录执行 `mysqld_safe` 应满足该条件(例如,二进制分发版为 `/usr/local/mysql`)。

如果不能根据工作目录找到服务器和数据库, `mysqld_safe` 试图通过绝对路径对它们定位。典型位置为 `/usr/local/libexec` 和 `/usr/local/var`。实际位置由构建分

发版时配置的值确定如果 MySQL 安装到配置时指定的位置,它们应该是正确的。

因为 `mysqld_safe` 试图通过工作目录找到服务器和数据库,只要您从 MySQL 安装目录运行 `mysqld_safe`, 可以将 MySQL 二进制分发版安装到其它位置:

```
shell> cd mysql_installation_directory
shell> bin/mysqld_safe &
```

如果 `mysqld_safe` 失败,即使从 MySQL 安装目录调用仍然失败,您可以指定 `--ledir` 和 `--datadir` 选项来指示服务器和数据库在您的系统中的安装目录。

一般情况,您不应编辑 `mysqld_safe` 脚本。相反,应使用命令行选项或 `my.cnf` 选项文件的 `[mysqld_safe]` 部分的选项来配置 `mysqld_safe`。一般不需要编辑 `mysqld_safe` 来正确启动服务器。

但是,如果您编辑,将来升级 MySQL 后会覆盖您修改的 `mysqld_safe` 版本,因此您应对您修改的版本进行备份以便将来重装。

在 NetWare 中, `mysqld_safe` 是一个 NetWare Loadable Module (NLM), 从原 Unix shell 脚本移植。它执行:

- 1) 检查系统和选项。
- 2) 检查 MyISAM 表。
- 3) 保持 MySQL 服务器窗口。
- 4) 启动并监视 `mysqld`, 如果因错误终止则重启。
- 5) 将 `mysqld` 的错误消息发送到数据目录中的 `host_name.err` 文件。
- 6) 将 `mysqld_safe` 的屏幕输出发送到数据目录中的 `host_name.safe` 文件。

15.3 MYSQL 程序概述

15.3.1 MYSQL 程序概述

MySQL AB 提供了几种类型的程序:

MYSQL 服务器和服务器启动脚本:

- 1) `mysqld` 是 MySQL 服务器。
- 2) `mysqld_safe`、`mysql.server` 和 `mysqld_multi` 是服务器启动脚本。
- 3) `mysql_install_db` 初始化数据目录和初始数据库。

访问服务器的客户程序:

- 1) `mysql` 是一个命令行客户程序,用于交互式或以批处理模式执行 SQL 语

句。

- 2) `mysqladmin` 是用于管理功能的客户程序。
- 3) `mysqlcheck` 执行表维护操作。
- 4) `mysqldump` 和 `mysqlhotcopy` 负责数据库备份。
- 5) `mysqlimport` 导入数据文件。
- 6) `mysqlshow` 显示信息数据库和表的相关信息。

独立于服务器操作的工具程序：

- 1) `myisamchk` 执行表维护操作。
- 2) `myisampack` 产生压缩、只读的表。
- 3) `mysqlbinlog` 是处理二进制日志文件的实用工具。
- 4) `pererror` 显示错误代码的含义。

大多数 MySQL 分发包括上述的全部程序，只是不包含那些与平台相关的程序。(例如，在 Windows 中不使用服务器启动脚本)。不同的只是 RPM 分发更加具体化。可能一个 RPM 是服务器分发，另一个 RPM 是客户程序分发等等。

15.3.2 调用 MYSQL 程序

要想从命令行调用 MySQL 程序(即从 shell 或命令提示)，应输入程序名，并随后输入指导操作发的选项或其它参量。下面的命令显示了一些程序调用的例子。`shell>`表示命令解释符提示；并不是输入的部分。您所看见的具体提示取决于命令解释符。典型提示符：`sh` 或 `bash` 为 `$`，`csh` 或 `tcsh` 为 `%`，Windows `command.com` 或 `cmd.exe` 为 `C:\>`。

```
shell> mysql test
shell> mysqladmin extended-status variables
shell> mysqlshow --help
shell> mysqldump --user=root personnel
```

以破折号开始的参数为选项参量。它们通常指定程序与服务器的连接类型或影响其操作模式。

非选项参数(不以破折号开始的参量)可以为程序提供更详细的信息。例如，`mysql` 程序将第一个非选项参量解释为数据库名，因此命令 `mysql test` 表示您想要使用 `test` 数据库。

后面的章节描述了具体的程序，表示程序可以理解的选项，并描述了其它非

选项参量的含义。

部分选项对部分程序是通用的。最常用的是指定连接参数的 `--host`、`--user` 和 `--password` 选项。它们指示 MySQL 服务器运行的主机和 MySQL 账户的用户名和密码。所有 MySQL 客户程序可以理解这些选项；它们允许您指定连接哪个服务器，以及在该服务器上使用的账户。

您也许会发现需要使用 MySQL 程序安装的 `bin` 目录的路径名来调用 MySQL 程序。如果您试图不在 `bin` 目录下运行 MySQL 程序，可能会遇到“程序未找到”错误。为了方便地使用 MySQL，可以将 `bin` 目录的路径名添加到 `PATH` 环境变量中。然后运行程序时只需要输入程序名，而不需要输入整个路径名。

关于设置 `PATH` 的指令的命令解释符请查阅相关文档。设置环境变量的语法与解释符有关。

15.3.3 指定程序选项

可以用几种方式提供 MySQL 程序的选项：

- 1) 在命令行中在程序名后面提供。这对于具体程序调用时使用的选项很普遍。
- 2) 在程序启动时读取的选项文件中设置。这对于每次程序运行时使用的选项很普遍。
- 3) 在环境变量中设置。这对每次程序运行时所使用的选项很有用，尽管实际上最常用选项文件。

MySQL 程序首先检查环境变量，然后检查选项文件，然后检查命令行来确定给出了哪些选项。如果多次指定一个选项，最后出现的选项占先。这说明环境变量具有最低的优先级，命令行选项具有最高优先级。

可以在选项文件中指定程序选项的默认值来让 MySQL 程序处理各选项。不需要在每次运行程序时输入选项，但可以根据需要通过命令行选项来覆盖默认值。

15.3.3.1 在命令行上使用选项

在命令行中指定的程序选项遵从下述规则：

- 1) 在命令名后面紧跟选项。
- 2) 选项参量以一个和两个破折号开始，取决于它具有短名还是长名。许多

选项有两种形式。例如，`-?`和`--help`是指导 MySQL 程序显示帮助消息的选项的短名和长名。

- 3) 选项名对大小写敏感。`-v` 和 `-V` 均有效，但具有不同的含义。(它们是 `--verbose` 和 `--version` 选项的短名)。
- 4) 部分选项在选项名后面紧随选项值。例如，`-h localhost` 或 `--host=localhost` 表示客户程序的 MySQL 服务器主机。选项值可以告诉程序 MySQL 服务器运行的主机名。
- 5) 对于带选项值的长选项，通过一个“=”将选项名和值隔离开来。对于带选项值的短选项，选项值可以紧随选项字母后面，或者二者之间可以用一个空格隔开。(`-hlocalhost` 和 `-h localhost` 是等效的)。该规则的例外情况是指定 MySQL 密码的选项。该选项的形式可以为 `--password=pass_val` 或 `--password`。在后一种情况(未给出密码值)，程序将提示输入密码。也可以给出密码选项，短形式为 `-ppass_val` 或 `-p`。然而，对于短形式，如果给出了密码值，必须紧跟在选项后面，中间不能插入空格。这样要求的原因是如果选项后面有空格，程序没有办法来告知后面的参量是密码值还是其它某种参量。因此，下面两个命令的含义完全不同：

a) `shell> mysql -ptest`

b) `shell> mysql -p test`

第一个命令让 `mysql` 使用密码 `test`，但没有指定默认数据库。第二个命令让 `mysql` 提示输入密码并使用 `test` 作为默认数据库。

部分选项控制可以开关的行为。例如，`mysql` 客户端支持 `--column-names` 选项，确定是否在查询结果开头显示一行栏目名。默认情况，该选项被启用。但是可能在某些情况下您想要禁用它，例如将 `mysql` 的输出发送到另一个只希望看到数据而不希望看到开始的标题行的程序中。

要想禁用列名，可以使用下面的形式来指定选项：

- 1) `--disable-column-names`
- 2) `--skip-column-names`
- 3) `--column-names=0`

4) `--disable` 和 `--skip` 前缀与 `=0` 后缀的效果相同：它们均关闭选项。

可以用下述方法启用选项：

- 1) `--column-names`
- 2) `--enable-column-names`
- 3) `--column-names=1`

如果选项有前缀 `--loose`，如果程序未识别出选项不会提示错误退出，但是会发出一条警告：

```
shell> mysql --loose-no-such-option
mysql: WARNING: unknown option '--no-such-option'
```

当您从安装了多个 MySQL 的同一台机器上运行程序时，`--loose` 前缀会很有用。当您在一个选项文件中列出选项时，该前缀会特别有用。有可能不能被程序的所有版本识别的选项可以冠以 `--loose` 前缀(或在选项文件中用 `loose`)。不能识别选项的程序版本将会发出一条警告并忽视该选项。

对 `mysql` 偶尔有用的另一个选项是 `-e` 或 `--execute` 选项，用来将 SQL 语句传递给服务器。该语句必须用引号引起来(单引号或双引号)。(然而，如果想要在查询中将值引起来，则对于查询应使用双引号，查询中引用的值应使用单引号)。当使用该选项时，语句被执行，然后 `mysql` 立即退出命令外壳。

例如，您可以用下面的命令获得用户账户列表：

```
shell> mysql -u root -p -e "SELECT User, Host FROM User" mysql
Enter password: *****
+-----+-----+
| User | Host |
+-----+-----+
|| gigan |
| root | gigan |
|| localhost |
| jon | localhost |
| root | localhost |
+-----+-----+
shell>
```

请注意 `mysql` 数据库名作为一个独立的参量传递。然而，相同的查询可能已

经使用 `mysql -u root -p -e "SELECT User, Host FROM mysql.User"` 从外壳中执行。

可以按这种方式传递多个 SQL 语句，用分号隔开：

```
shell> mysql -u root -p --execute="SELECT Name FROM Country WHERE Name LIKE
'AU%';SELECT COUNT(*) FROM City" world
Enter password: *****
+-----+
| Name |
+-----+
| Australia |
| Austria |
+-----+
+-----+
| COUNT(*) |
+-----+
| 4079 |
+-----+
```

请注意长形式(--execute)后面必须紧跟一个等号(=)。

-e 选项也可以以类似方式用来将命令传递给 MySQL 簇的 `ndb_mgm` 管理客户端。

15.3.3.2 使用选项文件

MySQL 程序可以从选项文件(有时也称为配置文件)读取启动选项。选项文件提供了一种很方便的方式来指定常用的选项，因此不需要每次运行程序时从命令行输入。

下面的程序支持选项文件：`myisamchk`、`myisampack`、`mysql`、`mysql.server`、`mysqladmin`、`mysqlbinlog`、`mysqlcc`、`mysqlcheck`、`mysqld_safe`、`mysqldump`、`mysqld`、`mysqlhotcopy`、`mysqlimport` 和 `mysqlshow`。

在 Unix 中，MySQL 程序从下面的文件读取启动选项：

表格 15-3 文件读取

文件名	目的
<code>/etc/my.cnf</code>	全局选项
<code>\$MYSQL_HOME/my.cnf</code>	服务器相关选项

defaults-extra-file	用--defaults-extra-file=path 指定的文件，如果有
~/my.cnf	用户相关选项

MYSQL_HOME 是一个环境变量，包含服务器相关的 my.cnf 文件驻留的目录路径。

如果未设置 MYSQL_HOME，并且 DATADIR 中有一个 my.cnf 文件，BASEDIR 中没有 my.cnf 文件，mysqld_safe 将 MYSQL_HOME 设置为 DATADIR。否则，如果未设置 MYSQL_HOME 并且在 DATADIR 中没有 my.cnf，则 mysqld_safe 将 MYSQL_HOME 设置为 BASEDIR。

典型情况二进制安装的目录为/usr/local/mysql/data 或源代码安装的目录为/usr/local/var。请注意这是配置时指定的数据目录的位置，而不是 mysqld 启动时使用--datadir 指定的。运行时使用--datadir 对寻找选项文件的服务器没有效果，因为服务器在处理命令行参量之前寻找这些选项。

MySQL 按照上述顺序寻找选项文件，并读取存在的选项文件。如果您想要使用的某个选项文件不存在，则用明文文本编辑器创建。如果存在多个选项文件，文件中指定的后读取的选项要优先文件中指定的先读取的选项。

 注释：在 Unix 平台上，MySQL 忽略人人可写的配置文件。这是故意的，是一个安全措施。

任何可以在运行 MySQL 程序时在命令行给出的长选项也可以在选项文件中给出。要想列出程序的适用选项，用--help 选项运行程序。

在选项文件中指定选项的语法类似于命令行语法，例外的是要忽略掉两个破折号。例如，命令行中的--quick 或--host=localhost 在选项文件中应指定为 quick 或 host=localhost。要想在选项文件中指定--loose-opt_name 形式的选项，应写为 loose-opt_name。

选项文件中的空行被忽略掉。非空行可以采用下面任何形式：

1) #注释；注释

注释行以“#”或“；”开头。“#”注释也可以从行的中部开始。

2) [group]

group 是您想要设置选项的程序名或组名。在组行后面，任何 opt_name 或 set-variable 行适用于组名，直到选项文件结尾或给出其它组行。

3) opt_name

等价于命令行中的--opt_name。

4) opt_name=value

等价于命令行中的--opt_name=value。在选项文件中，“=”字符附近可以有空格，而在命令行中是不允许的。您可以用单引号或双引号来引用值。如果值包含一个“#”注释字符或空格时很有用。

选项名和值前后的空白将自动删除掉。您可以在选项值中使用转义序列“\b”、“\t”、“\n”、“\r”、“\\”以及“\s”来表示退格、tab、换行符、回车以及空格字符。

所有客户程序(但不能被 `mysqld`)读取[client]选项组。这样允许您指定适用于所有客户端的选项。例如，[client]是用于指定连接服务器的密码的理想的组。(但应确保该选项文件只能由您自己读写，以便其他人不能发现您的密码)。一定不要随意在[client]组内放置选项，除非它可以被您使用的所有客户程序识别。如果您试图运行程序，如果程序不理解选项则会显示一条错误消息后退出。

从 5.0 系列的 MySQL 5.0.4 开始，可以在选项文件中使用!include 指令来包括具体文件和!includedir 来搜索选项文件的具体目录。例如，要包括文件 /home/mydir/myopt.cnf，可以使用：

```
!include /home/me/myopt.cnf
```

要搜索所有以.cnf 结尾的文件的目录/home/mydir 并作为选项文件读取，应使用：

```
!includedir /home/mydir
```

请注意这些选项与节有关。例如，假定您想要使用 my.cnf 中的某些内容，如下所示：

```
[mysqld]
!include /home/mydir/myopt.cnf
```

在这种情况下，只为该服务器处理文件 myopt.cnf，并且!include 指令将被客户应用程序忽略。然而，如果您使用下面的部分：

```
[mysqldump]
!includedir /home/mydir/my-dump-option
```

则只有 `mysqldump` 为以.cnf 结尾的文件检查目录/home/mydir/my-dump-option，服务器或其它客户应用程序均不检查。

如果您想要创建只由一个具体 `mysqld` 服务器发布系列读取的选项组，选项组可以用[mysqld-5.0]、[mysqld-5.1]等名称。下面的组表示--new 选项只能用于 5.1.x 版本的 MySQL 服务器：

```
[mysqld-5.1]
new
```

下面是一个典型的全局选项文件：

```
[client]
port=3306
socket=/tmp/mysql.sock
[mysqld]
port=3306
socket=/tmp/mysql.sock
key_buffer_size=16M
max_allowed_packet=8M
[mysqldump]
quick
```

在上述的选项文件中，设置 `key_buffer_size` 和 `max_allowed_packet` 变量的行使用了 `var_name=value` 语法。

下面是一个典型的用户选项文件：

```
[client]
# The following password will be sent to all standard MySQL clients
password="my_password"
[mysql]
no-auto-rehash
connect_timeout=2
[mysqlhotcopy]
interactive-timeout
```

所有支持选项文件的 MySQL 程序可以处理下面的命令行选项：

```
--no-defaults
```

不读取任何选项文件。

```
--print-defaults
```

打印从选项文件中获得的程序名和所有选项。

```
--defaults-file=path_name
```

只使用给出的选项文件。`path_name` 是文件的全路径名。

```
--defaults-extra-file=path_name
```

在全局选项文件后但在用户选项文件前读该选项文件。`path_name` 是文件的全路径名。

为了正确工作，每个选项必须紧随命令行中的命令名后，例外情况是 `--print-defaults` 可以紧随 `--defaults-file` 或 `--defaults-extra-file`。

在 shell 脚本中，可以使用 `my_print_defaults` 程序来分析选项文件。下面的例子显示了当要求显示[client]和[mysql]组内发现的选项时 `my_print_defaults` 产生的输出：

```
shell> my_print_defaults client mysql
--port=3306
--socket=/tmp/mysql.sock
--no-auto-rehash
```

15.3.3.3 用环境变量指定选项

要想用环境变量指定选项，使用适用注释处理器的语法来设置变量。例如，在 Windows 或 NetWare 中，可以设置 `USER` 变量来指定 MySQL 账户名。要想实现，使用语法：

```
SET USER=your_name
```

在 Unix 中的语法取决于您的外壳。假定您想要使用 `MYSQL_TCP_PORT` 变

量指定 TCP/IP 端口号。典型的语法为(例如 sh、bash、zsh 等等):

```
MYSQL_TCP_PORT=3306
export MYSQL_TCP_PORT
```

第 1 个命令设置变量，**export** 命令将变量导出到外壳环境，以便其值可供 MySQL 和其它进程访问。

csh 和 tcsh 有类似的问题。运行这些外壳时，使用 **setenv** 使外壳变量适用环境：

```
setenv MYSQL_TCP_PORT 3306
```

可以在命令提示符下执行设置环境变量的命令，以便立即生效。这些设定值持续到您注销。要向让这些设定值在您每次登录时生效，将相应命令放入每次启动时命令解释符所读的启动文件中。在 Windows 中典型启动文件为 AUTOEXEC.BAT，bash 为.bash_profile，或者 tcsh 为.tcshrc。

15.3.3.4 使用选项设置程序变量

许多 MySQL 程序有一些内部变量可以在运行时设置。程序变量的设置与带有值的其它长选项相同。例如，mysql 有一个 **max_allowed_packet** 变量，可以控制其通信缓冲区的最大长度。要想为 mysql 将 **max_allowed_packet** 变量的值设置为 16MB，使用下面的任何一个命令：

```
shell> mysql --max_allowed_packet=16777216
shell> mysql --max_allowed_packet=16M
```

第 1 个命令以字节指定值。第 2 个命令以兆字节指定值。变量值可以有一个后缀 K、M 或者 G(可以为大写或小写)来表示千字节、兆字节或者十亿字节的单位。

在选项文件中，变量设定值没有引导破折号：

```
[mysql]
max_allowed_packet=16777216
```

或：

```
[mysql]
max_allowed_packet=16M
```

16 POSTGRESQL 配置和使用

16.1 基本体系概念

先简单介绍 PostgreSQL 系统的基本体系，以理解 PostgreSQL 的部件之间的相互关系。PostgreSQL 使用客户端/服务器的模式，一次 PostgreSQL 会话由下列相关的进程(程序)组成：

一个服务器进程 —— 它管理数据库文件，接受来自客户端应用与数据库的联接，并且代表客户端在数据库上执行操作。数据库服务器程序叫做 `postmaster`。

需要执行数据库操作的用户的客户端（前端）应用 —— 客户端应用可能本身就是多种多样的：它们可以是一个字符界面的工具，也可以是一个图形界面的应用，或者是一个通过访问数据库来显示网页的 `web` 服务器，或者是一个特殊的数据库管理工具。一些客户端应用是和 PostgreSQL 一起提供的，但绝大部分是用户开发的。

和典型的客户端/服务器应用（C/S 应用）一样，这些客户端和服务端可以在不同的主机上，这时它们通过 `TCP/IP` 网络联接通讯。需要注意的是，在客户机上可以访问的文件未必能够在数据库服务器机器上访问（或者只能用不同的文件名进行访问）。

PostgreSQL 服务器可以处理来自客户端的多个并发请求。因此，它为每个请求启动一个新的进程。从这个时候开始，客户端和新服务器进程就不再经过最初的 `postmaster` 进程的干涉进行通讯。因此，`postmaster` 总是在运行，等待着联接，而客户端和相关联的服务器进程则是起起停停。

16.2 创建一个数据库

看看您能否访问数据库服务器的第一个例子就是试着创建一个数据库。一台运行着的 PostgreSQL 服务器可以管理许多数据库。通常我们会为每个项目和每个用户单独使用一个数据库。

要创建一个新的数据库（在我们这个例子里叫 `mydb`），可以使用下面的命令：

```
$ createdb mydb
```

它应该生成下面这样的响应：`CREATE DATABASE` 如果这样，那么这一步

就成功了。

如果您看到类似下面这样的信息：

```
createdb: command not found
```

那么就是 PostgreSQL 没有安装好；要么是就根本没装上，要么是您的搜索路径没有设置正确；尝试用绝对路径调用该命令试试：

```
$ /usr/local/pgsql/bin/createdb mydb
```

在您的节点上这个路径可能不一样，和您的管理员联系或者看看安装指导获取正确的位置。在中标麒麟高级服务器操作系统中的安装位置为 /usr/bin/createdb。

另外一种响应可能是这样：

```
psql: could not connect to server: Connection refused Is the server running locally and accepting
connections on Unix domain socket "/tmp/.s.PGSQL.5432"? createdb: database creation failed
```

这意味着该服务器没有启动，或者没有在 createdb 预期的地方启动。同样，您也要检查安装指导或者找管理员。

如果您没有创建数据库所需要的权限，那么您会看到下面的文字：

```
ERROR: CREATE DATABASE: permission denied createdb: database creation failed
```

并非所有用户都经过了创建新数据库的授权，如果 PostgreSQL 拒绝为您创建数据库，那么您需要让节点管理员赋予您创建数据库的权限。出现这种情况时请咨询您的节点管理员。如果是您自己安装了 PostgreSQL，那么您应该以您启动数据库服务器的用户身份登录然后参考手册完成权限的赋予工作。（注解：PostgreSQL 用户名是和操作系统用户账号分开的，如果您与一个数据库联接，您可以选择以何种 PostgreSQL 用户名进行联接；如果您不选择，那么缺省就是您的当前操作系统账号；如果这样，那么总有一个与操作系统用户同名的 PostgreSQL 用户账号用于启动服务器，并且通常这个用户都有创建数据库的权限；如果您不想以该用户身份登录，那么您也可以在任何地方声明一个 -U 选项以选择一个联接的 PostgreSQL 用户名）

您还可以用其它名字创建数据库，PostgreSQL 允许在一个节点上创建任意数量的数据库，数据库名必须是以字母开头并且小于 31 个字符长。一个方便的做法是创建和您当前用户名同名的数据库。许多工具假设该数据库名为缺省数据库名，所以这样可以节省您的敲键。要创建这样的数据库，只需要键入：

```
$ createdb
```

如果您再也不想使用您的数据库了，那么您可以删除它。比如，如果您是数据库 mydb 的所有人(创建人)，那么您就可以用下面的命令删除它：

```
$ dropdb mydb
```

对于这条命令而言，数据库名不能缺省，必须声明它。

这个动作物理上将所有与该数据库相关的文件都删除并且不可取消，因此做这件事之前一定要想清楚。

16.3 访问数据库

一旦您创建了数据库，您就可以访问它：运行 PostgreSQL 交互的终端程序 psql，它允许交互地输入、编辑和执行 SQL 命令。

也可使用现有的图形前端工具，比如 PgAccess 或者 ApplixWare(通过 ODBC)创建和管理数据库。

为 mydb 数据库启动 psql：

```
$ psql mydb
```

如果您省略了数据库名字，那么它缺省就是您的用户账号名字。

在 psql 里，您会看到下面的欢迎信息：

```
Welcome to psql, the PostgreSQL interactive terminal. Type: \copyright for distribution terms \h
for help with SQL commands \? for help on internal slash commands \g or terminate with
semicolon to execute query \q to quit mydb→
```

最后一行也可能是 mydb = #。

这个提示符意味着您是数据库超级用户，最可能出现在您自己安装了 PostgreSQL 的情况下；作为超级用户意味着您不受访问控制的限制。

如果您启动 `psql` 时碰到了问题，那么回到前面的小节。诊断 `psql` 的方法和诊断 `createdb` 的方法很类似，如果后者能运行那么前者也应该能运行。

`psql` 打印出的最后一行是提示符，这时候您就可以敲入 SQL 查询到一个 `psql` 维护的工作区间中。例如：

```
mydb→ SELECT current_date; date ----- 2011-05-04 (1 row) mydb→ SELECT 2 +
2; ?column? ----- 4 (1 row)
```

`psql` 程序有一些不属于 SQL 命令的内部命令，它们以反斜杠开头，"`\`"。有些这种命令在欢迎信息中列出。比如，您可以用下面的命令获取各种 PostgreSQL SQL 命令的帮助语法：`mydb→\h`。

要退出 `psql`，键入 `mydb→\q`。

然后 `psql` 就会退出并且返回命令行 `shell`。（要获取更多有关内部命令的信息，您可以在 `psql` 提示符上键入`\?`）`psql` 的完整功能在参考手册中有文档。如果 PostgreSQL 安装正确，那么您还可以在操作系统的 `shell` 提示符上键入 `man psql` 来阅读该文档。

16.4 POSTGRESQL 的简单使用

本节提供一个如何使用 SQL 执行简单操作的概述，另会有许多相关的书目中有完整的讲述可供参考。

下面的例子也可以在 PostgreSQL 源代码发布里的目录 `src/tutorial/` 中找到，请参考该目录中的 `README` 文件获取如何使用它们的信息。开始这个例子，按照下面的操作进行：

```
$ cd ....src/tutorial $ psql -s mydb ... mydb→ \i basics.sql
```

`\i` 命令从指定的文件中读取命令。`-s` 选项把您置于单步模式，它在向服务器发送每个查询之前暂停。在本节使用的命令都在文件 `basics.sql` 中。

16.4.1 基础概念

PostgreSQL 是一种关系型数据库管理系统，这意味着它是一种用于管理那些以关系形式存储的数据的系统。关系实际上是表的数学称呼，现在，把数据存储在表里的概念已经快成为固有的常识，但是还有其它的一些方法用于组织数据库。在类 Unix 操作系统上的文件和目录就形成了一种层次数据库的例子，更现

代的发展是面向对象的数据库。

每个表都是一个命名的行的集合，每一行由一组相同的命名列组成。而且每一列都有一特定的类型。虽然每列在每行里的位置是固定的，但一定要记住 SQL 并未对行在表中的顺序做任何保证(但您可以对它们进行明确的排序进行显示)。

表组成数据库，一个由 PostgreSQL 服务器管理的数据库集合组成数据库集群。

16.4.2 创建新表

您可以通过声明表的名字和所有字段的名字及其类型来创建表：

```
CREATE TABLE weather ( city varchar(80), temp_lo int, -- low temperature temp_hi int, -- high temperature prcp real, -- precipitation date date );
```

您可以在 psql 里连换行符一起键入这些命令，psql 可以识别该命令直到分号结束。

您可以在 SQL 命令中自由使用空白(包括空格，tab，和换行符)，这就意味着您可以用和上面不同的对齐方式键入命令；两个划线("--")引入注释，任何跟在它后面的东西直到该行的结尾都被忽略；SQL 是对关键字和标识符大小写不敏感的语言，只有在标识符

用双引号引起时才能保留它们的大小写属性。

varchar(80)声明一个可以存储最长 80 个字符的任意字符串的数据类型；int 是普通的整数类型；real 是一种用于存储单精度浮点数的类型；date 类型应该可以自解释。

PostgreSQL 支持通常的 SQL 类型 int, smallint, real, doubleprecision, char(N), varchar(N), date, time, timestamp 和 interval，还支持其他的通用类型和丰富的几何类型。PostgreSQL 客户化为定制任意的用户定义的数据类型，因而类型名并不是语法关键字，除了 SQL92 标准要求支持的特例外。

下面的例子将保存城市和它们相关的地理位置：

```
CREATE TABLE cities ( name varchar(80), location point );
```

类型 point 就是一种 PostgreSQL 特有数据类型的例子。

最后，我们还要提到如果您不再需要某个表，或者您想创建一个不同的表，

那么您可以用下面的命令删除它：

```
DROP TABLE tablename;
```

16.4.3 向表中添加行

INSERT 用于向表中添加行：

```
INSERT INTO weather VALUES ('San Francisco', 46, 50, 0.25, '1994-11-27');
```

请注意所有数据类型都使用了相当明了的输入格式；那些不是简单数字值的常量必需用单引号(')包围，就象在例子里一样；date 字段实际上对可接收的格式相当灵活。

point 类型要求一个座标对作为输入，如下：

```
INSERT INTO cities VALUES ('San Francisco', '(-194.0, 53.0)');
```

到目前为止使用的语法要求记住字段的顺序，一个可选的语法允许您明确地列出字段：

```
INSERT INTO weather (city, temp_lo, temp_hi, prcp, date) VALUES ('San Francisco', 43, 57, 0.0, '1994-11-29');
```

如果需要，可以用另外一个顺序列出字段或者是忽略某些字段，也就是说，以未知的顺序：

```
INSERT INTO weather (date, city, temp_hi, temp_lo) VALUES ('1994-11-29', 'Hayward', 54, 37);
```

许多开发人员认为明确列出字段要比依赖隐含的顺序是更好的风格。

您还可以使用 COPY 从文本文件中装载大量数据，这样做通常更快，因为 COPY 命令就是为这类应用优化的，同时还有比 INSERT 少一些的灵活性。比如：

```
COPY weather FROM '/home/user/weather.txt';
```

这里源文件的文件名必须是后端服务器可访问的，而不是客户端可访问的，因为后端服务器直接读取文件。您可以在参考手册中读到更多有关 COPY 命令的信息。

16.4.4 查询一个表

要从一个表中检索数据就是查询这个表。SQL 的 SELECT 就是做这个用途的。该语句分为选择列表(列出要返回的字段部分), 表列表(列出从中检索数据的表的部分), 以及可选的条件(声明任意限制的部分)。比如, 要检索表 weather 的所有行, 键入:

```
SELECT * FROM weather;
```

这里*意思是“所有字段”, 而输出应该是:

```
city | temp_lo | temp_hi | prcp | date -----+-----+-----+-----+----- San
Francisco | 46 | 50 | 0.25 | 1994-11-27 San Francisco | 43 | 57 | 0 | 1994-11-29 Hayward | 37 | 54 ||
1994-11-29 (3 rows)
```

您可以在目标列表中声明任意表达式, 比如:

```
SELECT city, (temp_hi+temp_lo)/2 AS temp_avg, date FROM weather;
```

这样应该得出:

```
city | temp_avg | date -----+-----+----- San Francisco | 48 | 1994-11-27 San
Francisco | 50 | 1994-11-29 Hayward | 45 | 1994-11-29 (3 rows)
```

请注意这里的 AS 子句是如何给输出字段重新命名的。(它是可选的)

允许使用任意布尔操作符 (AND, OR, 和 NOT) 给查询施加条件。比如, 下面的查询检索旧金山的下雨天的天气:

```
SELECT * FROM weather WHERE city = 'San Francisco' AND prcp > 0.0; Result: city | temp_lo
| temp_hi | prcp | date -----+-----+-----+-----+----- San Francisco | 46 | 50 |
0.25 | 1994-11-27 (1 row)
```

最后再提醒一下, 您可以要求选出来的结果按照某种顺序排序, 并且消除重复的行输出。(为了避免混淆, 特别声明 DISTINCT 和 ORDER BY 可以独立使用)

```
SELECT DISTINCT city FROM weather ORDER BY city; city ----- Hayward San
Francisco (2 rows)
```

16.4.5 在表之间连接

到目前为止, 我们的查询一次只访问了一个表。查询可以一次访问多个表,

或者用某种方式访问一个表，而同时处理该表的多个行。一个同时访问同一个或者不同表的多个行的查询叫连接(join)查询。举例来说，比如您想列出所有天气记录以及这些记录相关的城市。要实现这个目标，我们需要拿 **weather** 表每行的 **city** 字段和 **cities** 表所有行的 **name** 字段进行比较，并选取那些这些数值相匹配的行。

注意：这里只是一个概念上的模型，实际的连接可以以更高效的方式执行，但这些是用户看不到的。

这个任务可以用下面的查询来实现：

```
SELECT * FROM weather, cities WHERE city = name; city | temp_lo | temp_hi | prcp | date |
name | location -----+-----+-----+-----+-----+----- San
Francisco | 46 | 50 | 0.25 | 1994-11-27 | San Francisco | (-194,53) San Francisco | 43 | 57 | 0 |
1994-11-29 | San Francisco | (-194,53) (2 rows)
```

观察结果集的两个方面：

没有城市 **Hayward** 的结果行。这是因为在 **cities** 表里面没有 **Hayward** 的匹配行，所以连接忽略 **weather** 表里的不匹配行。我们稍后将看到如何修补这个毛病。

有两个字段包含城市名字。这是正确的，因为 **weather** 和 **cities** 表的字段是接在一起的。不过，实际上我们不想要这些，因此您将可能希望明确列出输出字段而不是使用*：

```
SELECT city, temp_lo, temp_hi, prcp, date, location FROM weather, cities WHERE city = name;
```

因为这些字段的名称都不一样，所以分析器自动找出它们属于哪个表，但是在连接查询里使用字段全称是很好的风格：

```
SELECT weather.city, weather.temp_lo, weather.temp_hi, weather.prcp, weather.date,
cities.location FROM weather, cities WHERE cities.name = weather.city;
```

到目前为止，这种类型的连接查询也可以用下面这样的形式写出来：

```
SELECT * FROM weather INNER JOIN cities ON (weather.city = cities.name);
```

这个语法并非象上面那个那么常用。

现在我们将看看如何能把 **Hayward** 记录找回来。我们想让查询干的事是扫描 **weather** 表，并且对每一行都找出匹配的 **cities** 表里面的行。如果我们没有找

到匹配的行，那么我们需要一些"空值"代替 cities 表的字段。这种类型的查询叫外连接(outer join)。(我们在此之前看到的连接都是内部连接。)这样的命令看起来象这样：

```
SELECT * FROM weather LEFT OUTER JOIN cities ON (weather.city = cities.name); city |
temp_lo | temp_hi | prcp | date | name | location
-----+-----+-----+-----+-----+-----+----- Hayward | 37 | 54 ||
1994-11-29 || San Francisco | 46 | 50 | 0.25 | 1994-11-27 | San Francisco | (-194,53) San Francisco
| 43 | 57 | 0 | 1994-11-29 | San Francisco | (-194,53) (3 rows)
```

这个查询是一个左手边外连接(left outer join)因为在连接操作符(注：LEFT OUTER JOIN)左手边的表中的行在输出中至少要出现一次，而在右手边的行将只输出那些与左手边行有对应匹配的行；如果输出的左手边表的行没有对应匹配的右手边表的行，那么在右手边行的字段将填充空(NULL)。

我们也可以把一个表和自己连接起来，这叫做自连接。比如，假设我们想找出那些在其它天气记录的温度范围之外的天气记录，这样我们就需要拿 weather 表里每行的 temp_lo 和 temp_hi 字段与 weather 表里其它行的 temp_lo 和 temp_hi 字段进行比较。我们可以用下面的查询实现这个目标：

```
SELECT W1.city, W1.temp_lo AS low, W1.temp_hi AS high, W2.city, W2.temp_lo AS low,
W2.temp_hi AS high FROM weather W1, weather W2 WHERE W1.temp_lo < W2.temp_lo AND
W1.temp_hi > W2.temp_hi; city | low | high | city | low | high
-----+-----+-----+-----+-----+----- San Francisco | 43 | 57 | San Francisco | 46 | 50
Hayward | 37 | 54 | San Francisco | 46 | 50(2 rows)
```

在这里我们把 weather 表重新标记为 W1 和 W2 以区分连接的左手边和右手边。您还可以用这样的别名在其它查询里节约一些敲键，比如：

```
SELECT * FROM weather w, cities c WHERE w.city = c.name;
```

以后可能会经常碰到这样的缩写。

16.4.6 聚集函数

和大多数其它关系数据库产品一样，PostgreSQL 支持聚集函数，一个聚集函数从多个输入行中计算出一个结果。比如，我们有在一个行集合上计算 count(数目)，sum(和)，avg(均值)，max(最大值)和 min(最小值)的函数。比如，我们可以用下面的语句找出所有记录中低温中的最高温度：

```
SELECT max(temp_lo) FROM weather; max ----- 46 (1 row)
```

如果我们想知道该度数发生在哪个城市，我们可以用：

```
SELECT city FROM weather WHERE temp_lo = max(temp_lo); WRONG
```

不过这个方法不能运转，因为聚集 **max** 不能用于 **WHERE** 子句中。(存在这个限制是因为 **WHERE** 子句决定哪些行可以进入聚集阶段；因此它必需在聚集函数之前计算。)不过，我们通常都可以用其它方法实现我们的目的；这里就可以使用子查询：

```
SELECT city FROM weather WHERE temp_lo = (SELECT max(temp_lo) FROM weather); city
----- San Francisco (1 row)
```

这样做是 OK 的，因为子查询是一次独立的计算，它独立于外层的 **select** 计算出自己的聚集。聚集同样也常用于 **GROUP BY** 子句，比如，我们可以获取每个城市低温的最高值：

```
SELECT city, max(temp_lo) FROM weather GROUP BY city; city | max -----+-----
Hayward | 37 San Francisco | 46 (2 rows)
```

这样给我们每个城市一个输出；每个聚集结果都是在匹配该城市的行上面计算的；我们可以用 **HAVING** 过滤这些分组：

```
SELECT city, max(temp_lo) FROM weather GROUP BY city HAVING max(temp_lo) < 40; city |
max -----+----- Hayward | 37 (1 row)
```

这样就只给出那些 **temp_lo** 数值曾经有低于 40 度温度的城市。最后，如果我们只关心那些名字以"S"开头的城市，我们可以用：

```
SELECT city, max(temp_lo) FROM weather WHERE city LIKE 'S%' GROUP BY city HAVING
max(temp_lo) < 40;
```

理解聚集和 SQL 的 **WHERE** 以及 **HAVING** 子句之间的关系对我们非常重要。**WHERE** 和 **HAVING** 的基本区别如下：**WHERE** 在分组和聚集计算之前选取输入行(因此，它控制哪些行进入聚集计算)，而 **HAVING** 在分组和聚集之后选取分组的行。因此，**WHERE** 子句不能包含聚集函数；因为试图用聚集函数判断那些行输入给聚集运算是没有意义的。相反，**HAVING** 子句总是包含聚集函数。(严格

说来，您可以写不使用聚集的 **HAVING** 子句，但这样做只是白费劲；同样的条件可以更有效地用于 **WHERE** 阶段。)

通过观察我们可以发现，我们可以在 **WHERE** 里应用城市名称限制，因为它不需要聚集。这样比在 **HAVING** 里增加限制更加高效，因为我们避免了为那些未通过 **WHERE** 检查的行进行分组和聚集计算。

16.4.7 更新

您可以用 **UPDATE** 命令更新现有的行。假设您发现所有 11 月 28 日的温度计数都低了两度，那么您就可以用下面的方式更新数据：

```
UPDATE weather SET temp_hi = temp_hi - 2, temp_lo = temp_lo - 2 WHERE date >
'1994-11-28';
```

看看数据的新状态：

```
SELECT * FROM weather; city | temp_lo | temp_hi | prcp | date
-----+-----+-----+-----+----- San Francisco | 46 | 50 | 0.25 | 1994-11-27 San
Francisco | 41 | 55 | 0 | 1994-11-29 Hayward | 35 | 52 | | 1994-11-29 (3 rows)
```

删除：

假设您对 **Hayward** 的天气不再感兴趣，那么您可以用下面的方法把那些行从表中删除。删除是用 **DELETE** 命令执行的：

```
DELETE FROM weather WHERE city = 'Hayward';
```

所有属于 **Hayward** 的天气记录都将被删除。

```
SELECT * FROM weather; city | temp_lo | temp_hi | prcp | date
-----+-----+-----+-----+----- San Francisco | 46 | 50 | 0.25 | 1994-11-27 San
Francisco | 41 | 55 | 0 | 1994-11-29 (2 rows)
```

我们用下面形式的查询的时候一定要小心：

```
DELETE FROM tablename;
```

如果没有限定条件，**DELETE** 将从指定表中删除所有行，把它清空。做这些之前系统不会请求您的确认。