注意，对 ErrorLog 和 TransferLog 命令说明的应该注意的安全问题也适用于 PidFile 命令。

在 httpd.conf 文件中需要配置的最后一条命令是服务器的 Internet 主机名：

```
ServerName [host name]
```

通常，需要输入诸如 www.yourcompany.com 这样的主机名。然而，确保在这里输入的主机名有正确的域名服务器记录将它指向你的服务器机器。下面讨论的配置文件是 srm.conf 文件。

3.1.2 srm.conf

超星阅览器提醒您：
使用本复制品
请尊重知识产权！

srm.conf 是资源配置文件。用它来告诉服务器你想在 Web 站点上提供什么资源以及在哪里和如何提供它们。列表 3-2 显示了缺省的 srm.conf 文件。

列表 3-2 由 srm.conf-dist 复制的缺省 srm.conf

```
# With this document, you define the name space that users see
of your http
# server. This file also defines server settings that affect
how requests are
# serviced, and how results should be formatted.

# See the tutorial at http://www.apache.org/for
# more information.

# Originally by Rob McCool; Adapted for Apache

# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this
directory, but
# symbolic links and aliases may be used to point to other
locations.

DocumentRoot /usr/local/etc/httpd/htdocs

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

UserDir public_html

# DirectoryIndex: Name of the file or files to use as a pre-
written HTML
# directory index. Separate multiple entries with spaces.

DirectoryIndex index.html
```

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

FancyIndexing on

# AddIcon tells the server which icon to show for different
files or filename
# extensions

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

AddIconByType (TXT, /icons/text.gif) text/*
AddIconByType (IMG, /icons/image2.gif) image/*
AddIconByType (SND, /icons/sound2.gif) audio/*
AddIconByType (VID, /icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl .gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^

# DefaultIcon is which icon to show for files which do not have
an icon
# explicitly set.

DefaultIcon /icons/unknown.gif

# AddDescription allows you to place a short description after
```

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

```
a file in
# server-generated indexes.
# Format: AddDescription "description" filename

# ReadmeName is the name of the README file the server will
look for by
# default. Format: ReadmeName name
#
# The server will first look for name.html, include it if
found, and it will
# then look for name an include it as plaintext if found.
#
# HeaderName is the name of a file which should be prepended to
# directory indexes.

ReadmeName README
HeaderName HEADER

# IndexIgnore is a set of filenames which directory indexing
should ignore
# Format: IndexIgnore name1 name2...
IndexIgnore */.??* *~ *# */HEADER* */README* */RCS

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

AccessFileName .htaccess

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

DefaultType text/plain

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

AddEncoding x-compress Z
AddEncoding x-gzip gz

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

超星阅览器提醒您:
此为本复制品
请尊重作者之著作权!

```
# it can understand. Note that the suffix does not have to be
the same
# as the language keyword --those with documents in Polish
(whose
# net-stand language code is pl) may wish to use
"AddLanguage pl .po"
# to avoid the ambiguity with the common suffix for perl
scripts.

AddLanguage en .en
AddLanguage fr .fr
AddLanguage de .de
AddLanguage da .da
AddLanguage el .el
AddLanguage it .it

# LanguagePriority allows you to give precedence to some
languages
# in case of a tie during content negotiation.
# Just list the languages in decreasing order of preference.

LanguagePriority en fr de

# Redirect allows you to tell clients about documents which
used to exist in
# your server's namespace, but do not anymore. This allows you
to tell the
# clients where to look for the relocated document.
# Format: Redirect fakename url

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

# Note that if you include a trailing / on fakename then the
server will
# require it to be present in the URL. So "/icons" isn't
aliased in this
# example.

# Alias /icons/ /usr/local/etc/httpd/icons/

# ScriptAlias: This controls which directories contain server
scripts.
# Format: ScriptAlias fakename realname
```

超星阅读器提醒您：
使用本复制品
请尊重相关知识产权！

```
# ScriptAlias /cgi-gin/ /usr/local/etc/httpd/cgi-bin/

# If you want to use server side includes, or CGI outside
# ScriptAliased directories, uncomment the following lines.

# AddType allows you to tweak mime.types without actually
editing it. or to
# make certain files to be certain types.
# Format: AddType type/subtype ext1

# AddHandler allows you to map certain file extensions to
"handlers",
# actions unrelated to filetype.  These can be either built into
the server
# or added with the Action command (see below)
# Format: AddHandler action-name ext1

# To use CGI scripts:
# AddHandler cgi-script .cgi

# To use server-parsed HTML files
# AddType text/html .shtml
# AddHandler server-parsed .shtml

# Uncomment the following line to enable Apache's send-asis
HTTP file
# feature
# AddHandler send-as-is asis

# If you wish to use server-parsed imagemap files, use
# AddHandler imap-file map

# To enable type maps, you might want to use
# AddHandler type-map var

# Action lets you define media types that will execute a script
whenever
# a matching file is called. This eliminates the need for
repeated URL
# pathnames for oft-used CGI file processors.
# Format: Action media/type /cgi-script/location
# Format: Action handler-name /cgi-script/location

# MetaDir: specifies the name of the directory in which Apache
can find
# meta information files. These files contain additional HTTP
```

```
headers
# to include when sending the document

# MetaDir .web

# MetaSuffix: specifies the file name suffix for the file
containing the
# meta information.

# MetaSuffix .meta

# Customizable error response (Apache style)
# these come in three flavors
#
#  1) plain text
# ErrorDocument 500 "The server made a boo boo.
# n.b. the (") marks it as text, it does not get output
#
#  2) local redirects
# ErrorDocument 404 /missing.html
# to redirect to local url /missing.html
# ErrorDocument 404 /cgi-bin/missing_handler.pl
# n.b. can redirect to a script or a document using server-
side-includes.
#
#  3) external redirects
# ErrorDocument 02
http://some.other_server.com/subscription_info.html
#
```

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

在 srm.conf 中使用的配置命令将在下面各节中进行讨论。这些命令用来为主 Web 站点创建目录配置。

3.1.2.1　Web 目录配置

像所有其他 Web 服务器一样，Apache 需要知道将要保存 Web 页面的顶层目录路径。在典型的情况下，此目录被称为文档根目录。Apache 提供名为 DocumentRoot 的命令，可以用它来指定顶层 Web 目录的路径。

可使用下面的语法，此命令告诉服务器将所提供的目录作为所有文档的根目录：

```
DocumentRoot [fully qualified path to your Web document
directory]
```

对读者来说，这是要做出的非常重要的决定。例如，如果按照下面所示来设置该命令：

```
DocumentRoot /
```

那么，系统中所有文件都可以被 Web 服务器访问。当然，通过提供正确的文件权限设置，可以保护自己的文件，但是，将文档根目录设置为系统的实际根目录肯定会带来严

重的安全风险。相反，你需要将文档根目录指向文件系统的一个具体子目录。缺省设置如下所示：

```
DocumentRoot /usr/local/etc/httpd/htdocs
```

然而，更好的选择可能是为你的机构创建一个 Web 目录结构。对于多用户、多域的系统建议使用图 3-3 表示的 Web 目录结构。

正如在图中所看到的，选择创建一个名为/www 的分区，并且，在它下面为系统中建立的每个 Web 站点都提供相应子目录。例如，/www/www.mycompany.com/站点有三个子目录：public、stage 和 development。而这三个子目录中的每一个都还有两个子目录：htdocs 和 cgi-bin。htdocs 是文档根目录，而 cgi-bin 则被用于 CGI 脚本。因此，www.mycompany.com Web 站点的 DocumentRoot 设置是：

```
DocumentRoot /www/www.mycompany.com/public/htdocs
```

此目录结构的优点是它在一个分区（/www）下面保存所有的 Web 文档和应用程序。这样就易于备份操作。而且，当指定网络中另一台机器提供 Web 服务的情况下，通过网络文件系统（NFS），可以将该分区装配到不同系统上。

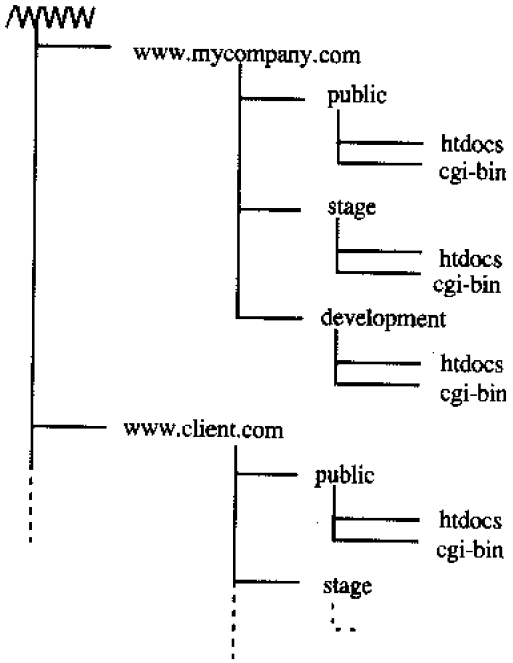

图 3-3 建议的 Web 目录结构

Cross-Reference 将在第 18 章中了解更多关于经过很好设计的 Web 目录结构。

注意，虽然文档根目录指向特定的目录，但这并不表示 Web 服务器不能访问在文档树之外的各个目录。通过使用符号链接（有正确的文件的权限设置）或使用别名（将在下一章讨论别名），可以很容易地使它做到这一点。从机构和安全的角度看，不推

荐使用大量符号链接或别名来访问在文档树之外的文件和目录。不管怎么说，有时在文档树外面保存一定类型的信息是必要的，虽然你需要保持服务器可以经常访问这一目录的内容。

使用本复制品
请尊重相关知识产权！

Tip　如果必须增加到文档树之外的其他目录的符号链接，确保在备份文件的时候，指导你的备份程序正确备份符号链接。

3.1.2.2　cgi-bin 目录配置

如果计划运行 CGI 脚本，将需要将这些脚本存储到 DocumentRoot 指定的目录（或它的子目录）以外的一个目录。由于安全的原因，你需要这样做。将脚本保存在文档树中，可能会使某些人能够看到你的脚本，并在其中找到弱点。这就是为什么在图 3-3 中的目录结构中包含名为 cgi-bin 的脚本目录的原因，而该目录位于每个站点的文档树的外面。

要将此目录保存在文档树外面，可以使用 ScriptAlias 命令来为该目录创建一个别名，如下所示：

```
ScriptAlias [/alias/] [fully qualified path for script
directory]
```

别名只是任意指定用来存储脚本、小程序以及其他内容的实际目录的名称。CGI 脚本目录的标准别名是 cgi-bin；然而，可以用自己喜欢的任何名称来自由地称呼它。缺省设置是：

```
ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-bin/
```

对于在图 3-3 中的 www.mycompany.com 站点，它的别名如下所示：

```
ScriptAlias /cgi-bin/ /www/www.mycompany.com/public/cgi-
bin/
```

因此，当要引用在 URL 中的脚本或在文档树中的 HTML 文件时，你应该使用别名，而不是实际文件的位置。例如：

```
http://www.mycompany.com/cgi-gin/finger.cgi
```

将请求在/www.mycompany.com/public/cgi-bin/目录中的 finger.cgi 脚本。考察此命令的另一种方法是：

```
ScriptAlias fakename realname
```

需要讨论的下一条命令是 UserDir 命令，它有如下语法：

```
UserDir [directory name]
```

用此命令来告诉 Apache，哪一个目录用作系统中用户的 DocumentRoot。只有在系统中拥有多个用户并允许每个用户有自己的 Web 目录时，此命令才有用。缺省的设置是：

```
UserDir public_html
```

它的含义是，如果将 Web 服务器的名称设置为 www.yourcompany.com，并且拥有两个用户（joe 和 jenny），那么它们的个人 Web 站点 URL 将如表 3-1 中所显示的那样。

表 3-1 用户的个人 Web 目录

URL	Web 目录
http://www.yourcompany.com/~joe	~joe/public_html
http://www.yourcompany.com/~jenny	~ jenny /public_html

注意，在 UNIX 系统中，~被扩展为用户的主目录。由 UserDir 命令指定的目录存在于每个用户的主目录中，而 Apached 对这一目录必须有读和执行的权限，以读取在 public_html 目录中的文件和子目录。在 UNIX 系统中，可使用如下命令做到这一点：

```
chown -R <user>.<Apache server's group name>
~<user>/<directory assigned in UserDir>
chmod -R 770 ~<user>/<directory assigned in UserDir>
```

例如，如果用户名是 joe，而 Apache 组被称为 httpd，并且在 UserDir 命令中将用户文档根目录指定为 public_html，那么上述命令应该像下面这样：

```
chown -R joe.httpd ~joe/public_html
chmod -R 2770 ~joe/public_html
```

第一个 chown 命令将~joe/public_html 目录（以及在它里面的全部文件和子目录）的所有者变为 joe.httpd。换句话说，它将在 public_html 目录中的全部文件和目录的全部所有权都给予用户 joe 和组 httpd。下一个 chmod 命令将访问权限设置为 2770——换句话说，只有用户(joe)和组(httpd)拥有在 public_html 中的全部读、写和执行的权限及在 public_html 下面的全部文件和子目录。它还确保当在 public_html 中创建新的文件或子目录时，新创建的文件设置了组 ID。这就允许 Web 服务器在没有用户介入的情况下也能访问此新文件。

Tip

如果使用脚本（如在 Linux 系统中的/usr/sbin/adduser 脚本）来创建用户帐号，你可能想要在此脚本中包含 Web 站点创建过程。只需要添加一个 mkdir 命令来创建一个缺省的 public_html 目录（如果那就是赋给 UserDir 命令的值），以创建 Web 目录。添加 chmod 和 chown 命令，来给予 Web 服务器用户读和执行在此公共目录下的文件和目录的权限。

需要设置的最后一条命令是 DirectoryIndex 命令，它有以下的语法：

```
DirectoryIndex [filename1, filename2, filename3,...]
```

此命令告诉 Apache 服务器应该将哪个文件作为被请求目录的索引。例如，当请求一个 URL，如 www.yourcompany.com/时，Apache 服务器认为此请求将访问 Web 站点的/（文档根）目录。如果 DocumentRoot 命令被设置为：

```
DocumentRoot /www/www.yourcompany.com/public/htdocs
```

那么，Apache 服务器将寻找一个名称为 www/www.yourcompany.com/public/htdocs/index.html 的文件。如果它找到该文件，Apache 通过将该文件内容返回给发出请求的 Web 浏览器来服务该请求。然而，如果 DirectoryIndex 被赋值为 welcome.html，而不是缺省的 index.html，则 Web 服务器将寻找 www/www.yourcompany.com/public/htdocs/welcome.

html。如果该文件不存在，Apache 通过创建动态 HTML 页面来返回目录列表。图 3-4 显示这种情况，在目录中不存在 index.html，服务器为发出请求的浏览器生成了目录列表。

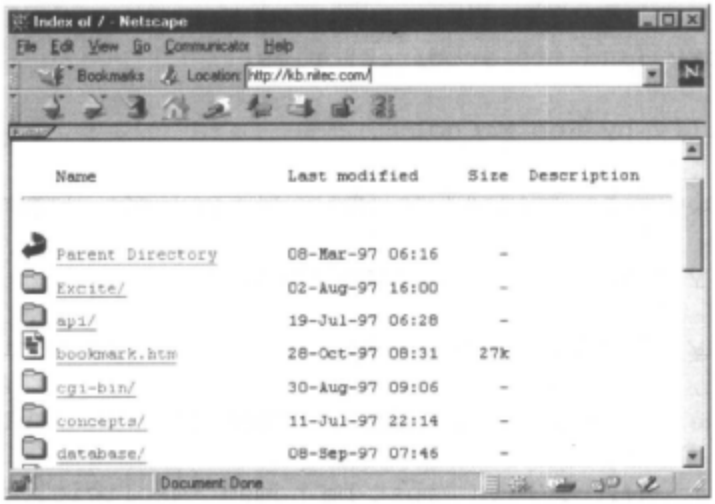

图 3-4　当不存在 index.html 时的动态目录列表

可以在 DirectoryIndex 命令中指定多个索引文件名。例如：

```
DirectoryIndex index.html index.htm welcome.htm
```

就告诉 Web 服务器，它应该检查这三个文件中任一个是否存在，如果找到其中的任一个，就将该文件返回给发出请求的 Web 客户。

注意，列出多个文件作为索引可能会引起两个问题。首先，服务器将不得不对每个目录请求检查大量文件是否存在，这有可能使它比平常慢一些。其次，将多个文件作为索引从机构的角度来看会使站点管理困难一些。然而，如果 Web 站点内容的开发人员使用不同的系统来创建文件，可将 index.html 和 index.htm 都作为索引文件，这样做可能是实际的解决方案。例如，一台 Windows 3.x 机器不能创建扩展名多于三个字符的文件名，因此，一个工作在这样一台机器上的用户可能需要用手工更改他（或她）在 Web 服务器上所有的 index.htm 文件。使用推荐的索引文件名就消除了这种烦恼。

3.1.3　access.conf

需要修改的最后一个配置文件是 access.conf。access.conf 文件被用来设置 Web 站点上诸如文件、目录和脚本项目的访问权限。列表 3-3 显示了缺省的 access.conf 文件。

列表 3-3　由 access.cong-dist 复制的缺省 access.conf

```
# access.conf: Global access configuration
# Online docs at http://www.apache.org/

# This file defines server settings which affect which types of
services
# are allowed, and in what circumstances.
```

```
# Allow server status reports, with the URL of
http://servername/server-status
# Change the ".your_domain.com" to match your domain to enable.

#<Location /server-status>
#SetHandler server-status

#order deny, allow
#deny from all
#allow from .your_domain.com
#</Location>

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

#<Location /cgi-bin/phf*>
#deny from all
#ErrorDocument 403 http://phf.apache.org/phf_abuse_log.cgi
#</Location>

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

这是仅有的需要在其中修改跨多行命令的配置文件。需要修改的第一条命令的语法如下：

```
<Directory directory> ... </Directory>
```

<Directory> 和 **</Directory>**被用来包容一组命令。被包容命令的作用范围被限于指名的目录（及子目录）。然而，只可以使用允许在目录范围内使用的命令（在下一章中将详细讨论这些命令）。

指名的目录或者是到目录的完整路径，或者是包含通配符的字符串。缺省的 access.conf 文件带有如下的命令设置：

```
<Directory /usr/local/etc/httpd/htdocs>
Options Indexes FollowSymLinks
AllowOverride None
order allow, deny
```

```
allow from all
 </Directory>
```

超星阅览器提醒您：
使用本复制品
请尊重相关法律知识产权！

你需要将目录/usr/local/etc/httpd/htdocs 改变为前面在 httpd.conf 文件中作为 DocumentRoot 命令参数设置的目录。缺省的设置包括诸如 Options 和 AllowOverride 之类的多条命令，它告诉服务器下面的内容：

♦指名的目录和它下面的所有子目录可以被索引。换句话说，如果有索引文件，将显示该文件；如果没有索引文件，服务器将为该目录创建动态索引。Options 命令指定这一内容。

♦指名的目录和它下面的所有子目录可以有符号链接，服务器可以沿着它们（也就是说，用作路径）来访问信息。Options 命令也指定了这一内容。

♦在 Directory 命令中指定的选项都不能被本地访问控制文件（由 srm.conf 中 AccessFileName 命令指定；缺省值为.htaccess）所覆盖。这是用 AllowOverride 命令指定的。

♦对所有用户都允许访问。

在开始阶段，缺省设置就应该足够了。然而，如果要将服务器连到 Internet 上，可能需要从 Options 命令行中删除 FollowSymLinks 选项。保留此选项会导致潜在的安全风险。例如，如果在 Web 站点中有一个目录没有索引页，服务器会显示自动产生的索引，它显示在该目录中可能有的全部符号链接。这会导致敏感信息被显示，或甚至可能允许任何人运行位于被不小心链接的目录中的可执行程序。

3.2 启动和停止服务器

现在，已经完成所有必须的基本配置，你已准备好运行服务器了。前面曾经说过，你可以用两种方法来运行服务器。让我们分别两种情况进行讨论。

3.2.1 独立的服务器

如果你设置服务器将它作为独立的服务器来运行，则下面的部分将告诉你在这种情况下如何运行、停止和重新启动服务器。

3.2.1.1 将 Apache 作为独立服务器运行

要运行独立服务器，只需要运行以前在系统中安装的 httpd 服务器程序。如果将 httpd 程序安装在/usr/sbin/目录下，那么以 root 用户登录，可以从任意位置使用如下的命令运行该程序：

```
/usr/sbin/httpd
```

这就启动 httpd 这个主 Web 服务器进程，它会在缺省的位置（在编译时指定的）寻找各个配置文件，如/usr/local/apache/conf/。如果将配置文件安装到了另一个目录，你需要使用下面的语句来告诉 httpd 到哪里去找到它们：

```
/usr/sbin/httpd -f /path/to/httpd.conf
```

其中/path/to 应该是到 httpd.conf 配置文件的实际路径。

在任何一种情况下，如果屏幕上没有错误信息，而且回到 shell 命令解释程序的提示符，该服务器就已成功地运行了。检查服务器是否在运行的最快方法是试用下面的命令：

```
ps auxw | grep httpd
```

它使用 ps 实用程序来列出在进程队列中的全部进程，并随后用管道将此输出传给 grep 程序。grep 在输出中搜索与关键字 httpd 匹配的行，并随后显示每个匹配行。如果看到了在其中带有关键字 root 的行，那就是你的主 Apache 服务器进程。注意，当服务器启动时，它创建一定数目的子进程来处理各个请求。如果作为 root 用户启动 Apache，父进程继续作为 root 运行，而子进程则变为在 httpd.conf 文件中指定的用户。

如果 httpd 给出不能“绑定”到一个地址的错误信息，原因将是：或者另一个进程正在使用已经配置给 Apache 使用的端口，或者你作为一般用户运行 httpd，却试图使用小于 1024 的端口（如缺省的端口 80）。

如果服务器不运行，请阅读运行 httpd 时出现的错误消息。还应该检查服务器的错误日志，以获得附加的信息（在缺省配置情况下，它在 logs 目录下的 error_log 文件中）。

如果想要服务器在系统重新引导后自动启动，你应该将一个对 httpd 的调用添加到系统启动文件中（典型情况下是 rc.local 或一个在 rc.N 目录中的文件）。可以将在列表 3-4 中的 httpd.sh 脚本用在一个启动文件中，使得在系统启动时或在一次重新引导后自动启动 Web 服务器。

列表 3-4　httpd.sh 脚本

```
#!/bin/sh
#
# httpd This shell script starts and stops the Apache server
# It takes an argument 'start' or 'stop' to receptively start
and
# stop the server process.
#
# Notes: You might have to change the path information used
# in the script to reflect your system's configuration.
#
[ -f /usr/sbin/httpd ] || exit 0

# See how the script was called.
case "$1" in
 start)
      # Start daemons.
     echo  -n "Starting httpd: "
     /usr/sbin/httpd
     touch  /var/lock/subsys/httpd
     echo
     ;;
  stop)
```

```
        # Stop daemons.
        echo -n "Shutting down httpd: "
        kill -TERM 'cat  /usr/local/etc/httpd/httpd.pid'

        echo "done"
        rm  -f /var/lock/subsys/httpd
        ;;
    *)
        echo  "Usage: httpd {start|stop}"
        exit 1
esac
exit  0
```

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

要启动 Apache 服务器，请按照下面所示运行该的脚本：

```
httpd.sh start
```

要停止 Apache 服务器，请按照下面所示运行该脚本：

```
httpd.sh stop
```

你需要将这些命令包含在系统中相应的 rc 文件里。你可以在 rc.local 文件中使用 start 命令，以在系统引导时自动启动服务器。

3.2.1.2 停止独立的 Apache 服务器

要停止独立的 Apache 服务器，可以使用 UNIX 的 kill 命令来发送 TERM 信号。TERM 信号告诉服务器程序（httpd）从容地结束它的运行。但是，在可以将 TERM 信号发送给主 Web 服务器进程之前，你需要知道正在作为 root 进程运行的 httpd 进程的进程 ID(PID)。

Apache 使得找到主 Web 服务器进程的 PID 更加方便。该 PID 被写入由 PidFile 命令指定的文件中。而且这一 PID 是用于主 httpd 进程的。不要试图杀死子进程，因为父进程将更新它们。停止服务器的典型命令是：

```
kill -TERM 'cat /usr/local/etc/apache/logs/httpd.pid'
```

如果你指定了不同的路径，而它的缺省值为 PidFile，那就使用被指定的值代替它。

3.2.1.3 重新启动独立的 Apache 服务器

如果你改变了一个或多个 Apache 配置文件，并需要运行中的服务器不结束该服务器进程而重新读取这些配置文件，所需要做的全部事情是使用 HUP 信号来重新启动主 Web 服务器进程，如下所示：

```
kill -HUP 'cat /usr/local/etc/apache/logs/httpd.pid'
```

这将强制主 Web 服务器进程重新读取配置文件。

3.2.2 由 inetd 运行的 Apache 服务器

如果你将服务器设置为通过 inetd 来运行，下面将告诉你如何运行、停止和重新启动服务器。

3.2.2.1 将 Apache 作为由 inetd 运行服务器

如果你为 httpd 正确地设置了 inetd 的配置，就不需要做任何事情来运行 Apache 服务器。因为 inetd 在接到访问 HTTP 端口的请求时，会自动运行 Apache 服务器。

3.2.2.2　停止由 inetd 运行的 Apache 服务器

通常情况下，你不需要结束由 inetd 运行的 Apache 服务器，因为它在服务请求之后立即消失。然而，如果由于某种原因，出现不寻常的情况（也许服务器被挂起了）不得不强制它退出。在这种情况下，你可以先确定该进程的 PID 如下：

```
ps  auxw | grep httpd
```

注意，在你自己的系统中，可能需要为 ps 实用程序使用不同的参数（有关细节请参考 ps 手册页）。第一条命令（ps）显示存在于你的系统进程队列中的所有进程，并将结果传给 grep 实用程序。grep 实用程序搜索与关键字 httpd 匹配的行，并显示所有匹配行。

现在，可以从 grep 的输出中确定进程的 PID。如果还不能确定，那就只运行此命令的 ps 部分，而在 ps 输出中的第一行应该会告诉你哪一列代表每个进程的 PID。一旦确定了 PID，只需要运行下面的命令：

```
kill -TERM [PID]
```

3.2.2.3　重新启动由 inetd 运行的 Apache 服务器

重新启动由 inetd 运行的服务器不是必要的。因为 inetd 将为每个请求启动新的服务器进程。如果刚刚更新了配置文件，则你或其他人发出的下一个请求将受到在配置文件中所做变动的影响。

3.3　测试 Apache 服务器

现在，已经准备好测试你运行的 Apache 服务器了。第一个测试是简单的。

运行自己选择的 Web 浏览器，并让它指向运行新配置好的 Apache 服务器的 Web 站点。如果在运行 Apache 的同一个系统中运行这一 Web 浏览器，可以使用下面的 URL：

```
http://localhost/
```

在所有其他情况下，需要指定确切的主机名称（如 www.yourcompany.com）。

如果尚未对缺省的 htdocs 目录做任何变动，你将看到如图 3-5 所示的页面。

此页面是和 Apache 发行版一起提供的，并需要用自己的内容进行改变。

最后，需要确保日志文件被正确地更新了。要检查日志文件，请输入日志的目录，并运行下面的 UNIX 命令：

```
tail -f [name of access log]
```

tail 是 UNIX 的实用程序，可以用来检查不断增长的文件（当指定 -f 选项时）。现在，使用 Web 浏览器访问此站点，或者如果你已经访问了该站点，只需要将你当前的页面重新加载到浏览器上。你应该在屏幕上看到一个条目添加到列表中。多按几次 reload（重新装载）按钮，以确保访问文件被相应地更新了。如果看到更新过的记录，那么你的访问日志文件就在工作了。按下 Control+C 键退出 tail 命令会话过程。如果在文件中没有看到任何新记录，你应该检查日志文件和保存日志文件的目录的权限设置。

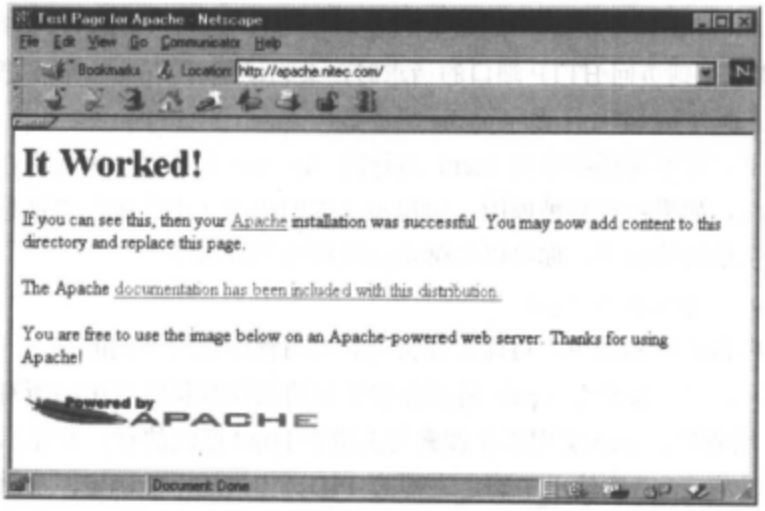

图 3-5 缺省的 Apache 主页

另一个要检查的日志是错误日志文件。使用：

```
tail -f [name of error log]
```

可看到加入的错误日志条目。只需要在你的 Web 浏览器上请求不存在的资源（如一个没有的文件），就将看到正被加入的条目。如果观察到了这一点，则错误日志文件就已被正确配置了。

如果所有这些测试都成功了，那么你就已经成功地配置了 Apache 服务器。祝贺你！

3.4 小　　结

在本章中，讨论了 Apache 的配置基础。我的目标是将细节减到最少，以便在短时间内准备好你的 Apache 服务器，并让它运行起来。讨论了如何将 Apache 作为独立的 Web 服务器或由 inetd 运行的服务器。并讨论了如何配置 httpd.conf、srm.conf 和 access.conf 文件。

下一章将讨论 Apache 的内核命令。

第4章 内 核 命 令

本章内容:

♦理解命令的适用范围

♦Apache 的内核命令

directive 就是作用于 Apache 的一条命令。Apache 从在第 3 章中讨论的配置文件中读取命令。通过使用命令，Apache 管理员可以控制 Web 服务器的行为。有很多命令可用于 Apache，这就使得 Apache 成为高度可配置的 Web 服务器。作为基本 Apache 安装的一部分命令称为内核命令。这些命令总是可以使用的。还有一些由标准模块提供的其他命令。这些模块也是 Apache 标准发行版的一部分。

在第 5 章中，将讨论标准模块提供的命令。在本章中，讨论（内核的和所有其他的）命令标准的适用范围，同时还深入讲解内核命令。本章并不是对所有可能的内核命令提供按字母顺序的列表，而是根据它们的用法对它们进行分类组合。这些类型有：一般配置、性能和资源配置、标准容器、虚拟主机专用的配置、记录日志以及认证与安全。每条命令的说明都提供了如下的信息：

♦语法——说明命令的名称和它可能采用的所有参数或值。

♦使用范围——指定命令的适用范围（或范围）。

♦缺省值——此行说明命令的缺省值。只对适用的命令进行说明。

♦AllowOverride（允许覆盖）——用它指定的值来决定是否允许使用在针对目录的访问配置文件（缺省情况下是.htaccess）中的命令。只对适用的命令进行说明。

♦兼容性——此行告诉你命令适用的 Apache 版本的细节。

在涉及具体的内核命令之前，让我们先讨论命令的适用范围。

4.1 Apache 命令的适用范围

在确定使用任何内核命令之前，了解一条命令的适用范围是很重要的；换句话说，需要知道命令的使用范围。一旦理解了讨论内核命令所需要的术语，就将讨论内核命令本身。

使用命令有三种主要的范围。命令可以出现在主服务器配置文件中任何容器（它看起来很像 HTML 标记）的外面（服务器配置的使用范围）、在容器之内（容器的使用范围）或者在针对目录的配置文件中的任意位置（针对目录的使用范围）。

4.1.1 服务器配置的使用范围

这些命令可以出现在主服务器配置文件中任何容器外面的任何位置。可以认为这一使

用范围是全局的使用范围。在缺省情况下，适用此使用范围的命令会影响所有其他使用范围。一条经验规则是：没有包含在容器中的命令被认为是全局命令。这些命令可以用在服务器配置文件（例如 httpd.conf、srm.conf 和 access.conf）中的任何位置，但却不能用在任何容器或针对目录的配置文件（.htaccess）中。

4.1.2 容器的使用范围

要限制一条命令的使用范围，可以使用容器。一对容器标记封装一组命令，并因此将这些命令的范围限制在标记对的范围内。Apache 提供以下的标准容器：

♦<VirtualHost...>...</VirtualHost>被用来将一条或多条命令应用到在容器的打开标记中指定的虚拟主机。

♦<Directory...>...</Directory>被用来将一条或多条命令应用到指定的目录。注意，如果使用此容器标记来为目录指定一条或多条命令，这些命令也自动适用于所有的子目录。然而，如果这不是想要的副作用，你可以为每个子目录创建单独的目录容器，并在该目录的每层中以不同方法控制服务器的行为。

♦<DirectoryMatch regex>...</DirectoryMatch>与<Directory>容器完全相同，然而，它用的是规则表达式（regex），而不是一般的目录名作为参数。

Note

一个规则表达式一般是由正常字符和特殊字符组成，以创建一个模式。该模式被用来匹配一个或多个子字符串或一个完整的字符串。要知道更多关于规则表达式的内容，请参见附录 C。

♦<Files...>...</Files>容器被用来将一条或多条命令应用到一个特定文件或一组文件。

♦<FilesMatch regex>...</FilesMatch regex>与<Files>容器完全相同，然而，它使用规则表达式（regex），而不用一个或多个文件名作为参数。

♦<Location...>...</Location>容器被用来将一条或多条命令应用到特定 URL。

♦<LocationMatch regex>...</LocationMatch regex> 与<Location>容器完全相同，然而，它用规则表达式（regex），而不用 URI 作为参数。

♦<Limit...>...</Limit>容器被用来应用一条或多条命令来控制对 Web 站点的特定区域或特殊 HTTP 请求方法的访问。在所有容器中此类容器的作用范围最小。

Note

URI（统一资源标识）是统一资源标识系列的类的术语，而 URL 只是它的成员。其他成员有：统一资源名称（URN）、统一资源特征（URC）以及与位置无关的文件名称（LIFN）。然而，只有 URL 被广泛使用。

让我们看容器范围的例子。下面显示了 httpd.conf 文件的一段：

```
<VirtualHost  206.171.50.50>
ServerName  www.nitec.com
DocumentRoot  /www/nitec/public/hodocs
DirectoryIndex welcome.html
```

```
<Location /secured/>
        DirectoryIndex  login.html
</Location>
</VirtualHost>
```

在此例子中，使用<VirtualHost>容器定义称为 www.nitec.com 的虚拟主机。使用了三条在虚拟主机范围内使用的命令：ServerName、DocumentRoot 和 DirectoryIndex，因此，它们适用于整个虚拟主机。DirectoryIndex 命令指定：如果提出了访问此虚拟服务器的目录的请求（如果可以使用的话），将返回名为 welcome.html 的文件。然而，Location 容器则指定，当有人试图访问 www.nitec.com/secured/ URL 时，应该返回另一个文件 login.html。在较小范围容器中的命令总是覆盖在较大范围容器中的命令。

当使用任何容器来为 Web 空间的一部分定义行为时，应该使用如下的一些规则：

- <VirtualHost>容器不能被嵌套在任何其他种类的容器中。
- 在使用范围最小的<Limit>容器中不能有其他容器。
- <Files>容器中只能用最小的<Limit>容器。
- <Location>和<Directory>容器不混合，因此，不要在一个容器里使用另一个。

4.1.3 针对目录的使用范围

还可以在针对目录的配置文件中包含一些命令。一个针对目录的配置文件（缺省的文件名为.htaccess）是一个文本文件，它包含一个或多个只适用于当前目录的命令。这些命令也可以被包含在诸如<Files...>和<Limit...>这样的容器中。当提出对目录中的文件的请求时，通过使用此文件可以控制 Apache 如何行动。如果一条命令在此使用范围内有效，就表示它出现在针对目录的访问配置文件中。

注意，在服务器配置或虚拟主机范围内，禁止针对目录的配置文件中的全部或部分可以被覆盖的内容是可能的。因此，取决于当前使用的覆盖命令，在此使用范围内的所有命令可以不被处理。

4.2 一般的配置命令

从本质上讲这些命令是基础性的，通常适用于主服务器（服务器配置的使用范围）和虚拟服务器（虚拟主机的使用范围）。

4.2.1 AccessConfig

语　　法：AccessConfig filename

缺 省 值：AccessConfig conf/access.conf

使用范围：服务器配置，虚拟主机

在第三章中曾经说过，Apache 服务器在启动时读取一组配置文件。这些文件之一是称为 access.conf 的全局访问配置文件。在缺省情况下，此文件位于服务器根目录下的 conf

子目录中，而根目录是由 ServerRoot 命令指定的。

通过使用 AccessConfig 命令，可以指示 Apache 去读取不同于缺省 conf/access.conf 文件的另一个文件。例如，如果需要从称为 conf/globalaccess.conf 的文件中装入全局访问信息，需要在 httpd.conf 文件中使用如下的命令：

```
AccessConf conf/globalaccess.conf
```

Note　filename 必须是相对于 ServerRoot 的路径名称。

Tip　如果需要为每个运行 Apache 的服务器准备单独的配置文件，可以按如下所示设置 AccessConfig 来禁止装入 conf/access.conf 文件：

```
AccessConfig  /dev/null
```

还需要以相同的方式使用 ResourceConfig 命令来禁止装入 conf/srm.conf 文件。有关详细内容，请参见 ResourceConfig 命令。

前不久，access.conf 文件还只能包含<Directory>段；现在，它可以包含允许在服务器配置使用范围内的任何服务器命令。它在虚拟主机使用范围中也是合法的，因此，你可以使用它来从<VirtualHost>命令中装入配置文件。然而，在<VirtualHost>段中 AccessConfig 只允许使用一次。如果要支持多个虚拟主机，这可以带来方便。可以使用此命令为每个虚拟主机装入单独的配置文件，因而提高主服务器配置文件的可读性。

注意，Apache 先读取由 ResourceConfig 命令指定的文件再读取由 AccessConfig 命令指定的文件。

4.2.2 AccessFileName

语　法：`AccessFileName  filename filename ...`

缺 省 值：`AccessFileName .htaccess`

使用范围：服务器配置，虚拟主机

兼 容 性：在 Apache1.3 和以后的版本中可以使用一个以上的 filename 选项。

此命令指定针对目录的访问控制文件的名称。缺省的设置（.httaccess）使得 Apache 在客户系统每次发出访问请求时要查找.htaccess 文件。例如，假定名为 www. mycompany. com 的 Apache Web 站点将 DocumentRoot 命令设置为：

```
DocumentRoot  /www/mycompany/public/htdocs
```

而且， Web 浏览器请求如下的 URL：

```
www.mycompany.com/feedback.html
```

这将导致 Apache 去搜索下列的访问控制文件：

- /.htaccess
- /www/.htaccess
- /www/mycompany/.htaccess

- /www/mycompany/public/.htaccess
- /www/mycompany/public/htdocs/.htaccess

在 Apache 检查了所有这些文件之后，它会寻找 feedback.html 文件。看起来这有大量的磁盘 I/O 工作，它实际就是这样的！如果不使用针对目录的访问控制文件，并需要 Apache 停止检查该文件，只需要按如下所示使用<Directory>命令来禁止覆盖选项：

```
<Directory />
AllowOverride None
</Directory>
```

更多的细节请参见本章中有关<Directory>和 AllowOverride 命令的各节。

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

4.2.3 BindAddress

语　　法：BindAddress IP address

缺 省 值：BindAddress *

使用范围：服务器配置

运行在 UNIX 系统上的 Apache 可以在一个或多个 IP 地址上侦听连接请求。此命令使你能够指定 Apache 服务的 IP 地址。此命令的缺省值是*（通配符），它表示 Apache 侦听连接到服务器机器的所有 IP 地址。例如：

```
BindAddress 206.171.50.50
```

将使 Apache 只侦听指定的 IP 地址。只可以使用一个 BindAddress 命令。要更多地控制 Apache 侦听哪些地址和端口，请用 Listen 命令来代替 BindAddress。BindAddress 可以被用作使用多个独立服务器支持虚拟主机的备选方法，而不是使用<VirtualHost>段。

4.2.4 CoreDumpDirectory

语　　法：CoreDumpDirectory directory

缺 省 值：与 ServerRoot 指定的位置相同

使用范围：服务器配置

如果你能够检查内核转储文件，并想使 Apache 在发生故障时在特定的文件中转储内核，将此命令设置为存放内核转储文件的目录。它的缺省值是由 ServerRoot 指定的目录。然而，因为这一目录不应该由运行该服务器的用户写入，也不能在该目录下写内核转储文件。因此，缺省值很可能是错误的。

4.2.5 DocumentRoot

语　　法：DocumentRoot directory

缺 省 值：DocumentRoot /usr/local/apache/htdocs

使用范围：服务器配置，虚拟主机

由此命令指定的目录成为所有由 Apache 提供的文档的顶层目录。例如，如果为

www.mycompany.com 设置了：

```
DocumentRoot /www/mycompany/public/htdocs
```

那么对 www.mycompany.com/corporate.html 的请求将使服务器寻找如下的文件：

```
/www/mycompany/public/htdocs/corporate.html
```

如果找到了这一文件，将它返回给客户（也就是 Web 浏览器）。类似的，如果客户请求 www.mycompany.com/info/corporate.html，服务器寻找如下的文件：

```
/www/mycompany/public/htdocs/info/corporate.html
```

Caution 如果在 DocumentRoot 的参数后面带有斜杠（例如，DocumentRoot /usr/web/），在内核模块（mod_dir）中的隐错会导致问题。因此，应该避免在任何命令中的任何路径后面加入斜杠（/）字符。

注意，使服务器寻找在 DocumentRoot 目录以外的目录中的文件是可能的。换句话说，如果要访问在 DocumentRoot 树以外的文件，你可以使用 Alias 命令来创建一个虚目录名称，该名称可以指向在服务器文件系统中的任何一个实际目录。

Cross-Reference 关于 Alias 命令的详细内容，请参见第 5 章。

4.2.6 ErrorDocument

语　　法：ErrorDocument error-code filename | error message | URL
使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件
覆　　盖：FileInfo
兼 容 性：在 Apache 1.1 和以后的版本中，可用于目录和针对目录的配置文件（.htaccess）使用范围。

当服务器遇到问题时，它形成标准的错误消息，其中带有错误代码。然而，对大多数人来说，这不是友好的用户界面，因此，需要定制的错误消息或可能的恢复措施。如果需要定制的错误消息，使用 ErrorDocument 命令来覆盖标准的错误消息。

此命令需要两个参数。第一个参数是标准的 HTTP 错误代码，可以在附录 B 中找到错误代码，而第二个参数是对该错误的操作项。取决于你的需要，可以定义对特定错误条件需要服务器采取的操作。

例如，如果想为所有导致标准的“文件未找到”消息的请求提供定制消息，那么你需要做的全部事情就是找到对应该错误条件的服务器状态码，并使用 ErrorDocument 命令。“文件未找到”的服务器状态码是 404，下面的命令设置使得 Apache 能够显示定制的错误消息：

```
ErrorDocument 404 "Sorry, this is an invalid request
since %s"
```

注意，整个消息需要加引号。服务器会用可以得到的错误信息来代替%s。然而，如果你认为此方法有限制，可以使用文件来提供错误消息。例如：

```
ErrorDocument 404 /errors/404.html
```

无论何时发生了“文件不存在”的错误，服务器在 DocumentRoot 目录下的 errors 目录中找到 404.html 文件，并将它返回给客户（Web 浏览器）。如果你不想仅仅返回静态页面，可以使用 CGI 脚本来执行特定的操作。在这样的情况下，使用对 CGI 脚本的调用来代替文件名：

```
ErrorDocument 404 /cgi-bin/missingurl.cgi
```

在每次发生 404 错误时，它将调用名为 missingurl.cgi 的 CGI 脚本。你还可以不使用文件名而使用 URL，将客户重定向到另一个站点：

```
ErrorDocument 404 http://www.newsite.com/we.moved.html
```

当一个页面或整个站点已经移到另一个位置时，可以使用此命令。

Note 如果发生了 401（未经授权）错误，则不能将客户指向一个远程服务器。此命令的值必须是本地文件或消息。

4.2.7 Include

语　法：Include filename

使用范围：服务器配置

兼 容 性：可用在 Apache 1.3 和以后的版本中。

如果想要在服务器配置文件的里面包含其他配置文件，请使用此命令。

4.2.8 Listen

语　法：Listen [IP address:] port number

使用范围：服务器配置

在缺省情况下，Apache 响应所有连接到服务器机器的 IP 地址的请求，但却只响应由 Port 命令指定的端口地址。Listen 命令可以使这种情况更容易配置。可以使用 Listen 命令来告诉 Apache 对特定的 IP 地址、IP 地址和端口的组合或者只对端口本身作出响应。

尽管可以用 Listen 来代替 BindAddress 和 Port，但如果 Apache 服务器产生了指向它自己的 URL，你可能不得不使用 Port 命令。

可以使用多个 Listen 命令来指定要侦听的一定数目的地址和端口。服务器会响应任何一个来自列出的地址和端口的请求。例如，要使服务器接受在端口 80 和端口 8080 上的连接，使用：

```
Listen 80
Listen 8080
```

下面的例子使得 Apache 接受在两个 IP 地址和端口号上的连接：

```
Listen 11.22.3311.22.33.1:80
Listen 11.22.3311.22.33.2:8080
```

4.2.9 Port

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

语　　法：Port number
缺 省 值：Port 80
使用范围：服务器配置

此命令将范围在 0 到 65535 中的端口号指定给一个主机。在没有用任何 Listen 命令或 BindAddress 命令指定端口号的时候，Port 命令设置服务器侦听的网络端口。如果已有任何 Listen 或 BindAddress 命令指定了端口号，那么，Port 命令对服务器侦听的地址没有影响。Port 命令设置 SERVER_PORT 环境变量（供 CGI 和 SSI 使用），并在服务器必须产生指向它自己的 URL 时被使用。

尽管可以在 0 到 65535 之间指定端口号，但应该记住一个限制。小于 1024 的所有端口是为标准服务保留的，例如 TELNET、SMTP、POP3、HTTP 和 FTP。你可以在/etc/services 文件中找到所有指定给标准服务的端口号。或者，为了安全起见，对你的 Apache 服务器使用 80 以外的任何端口号（使用高端地址，例如 8000）。

Tip

如果你是非 root 用户，并为了实验或某种其他不重要的原因运行 Apache，你需要使用大于 1024 的端口，因为只有 root 用户才能在这些保留的端口上运行诸如 Apache 这样的服务。

Note

也可以用<VirtualHost>容器来设置用于虚拟主机的端口。

4.2.10 User

语　　法：User unix-userid
缺 省 值：User #-1
使用范围：服务器配置，虚拟主机

此命令设置用户 ID，由响应请求的子 Apache 服务器进程使用这一 ID。为了使用此命令，必须将 Apache 配置作为独立的服务器来运行（参见 ServerType 命令），而且开始必须由 root 运行。一旦启动了 Apache 主服务器，它就启动子进程来响应请求。然而，这些子进程不是由 root 运行的。主 Apache 进程（经常被称为守护进程）将子进程的用户 ID 变为在 User 命令中设置的 ID，只要它是合法的用户 ID。

Caution

如果你作为非 root 用户启动服务器，它不能将子进程的 ID 变为由 User 命令指定的用户 ID，相反将继续作为原来的用户运行。如果你确实作为 root 启动服务器，那么主 Apache 进程保持作为 root 运行；然而，它将子进程作为由 User 命令指定的用户来运行。

还可以使用用户 ID 编号，你通常可以在/etc/password 文件中找到这些编号。如果你计划使用编号，而不是实际的用户名称，在该编号的前面应该加一个#号。

很多Apache管理员为他们的站点使用缺省的nobody用户。此用户不是在所有的UNIX系统中都可以使用的，而且也不总是可取的。强烈建议你为 Apache 服务器采用单独的用户和组 ID（参见 Group 命令）。这样做将使你更好地控制服务器能访问和不能访问的范围。Apache 子进程使用的用户 ID 应该只有很少的访问权限。它应该不能访问不想被外部世界看到的文件。同样，此用户应该不能执行不打算用于 HTTP 请求的应用程序。

在<VirtualHost>容器中使用此命令需要一个正确配置的 suEXEC 的 wrapper 程序。当该程序被以此种方式用在<VirtualHost>容器内的时候，只有运行 CGI 的用户受到影响。非 CGI 请求仍然由主 User 命令中指定的用户处理。因此，主 User 命令并不完全被覆盖。

最后，绝不要将 User（或 Group）命令设置为 root，除非确切知道你正在做什么和它的危险是什么。

4.2.11 Group

语　　法：Group Unix-group

缺 省 值：Group #-1

使用范围：服务器配置，虚拟主机

Group 命令应该和 User 命令一起使用。Group 命令确定独立服务器回答请求时使用的组名。为了使用此命令，独立服务器最初始必须由 root 运行。Group 命令还可以用来指定组号。请在/etc/group 文件中查阅各个组名以及它们对应的组号。

Note　对 User 命令提出的所有警告和建议也适用于此命令。确保你阅读了 User 命令的细节。

4.2.12 <IfModule>

语　　法：<IfModule [!]module-name>... </IfModule>

缺 省 值：无

使用范围：全部

兼 容 性：只可用在 1.2 和以后的版本中。

如果你有一些由定制模块提供的命令，而这一模块并不始终包含在你的 Apache 安装中，应该使用此命令。例如，如果你希望只在可以使用特定模块时才使用它所提供的命令，那么，可以使用下面的条件结构：

```
<IfModule module-name>
# assign the following directives their respective value
# your directives goes here
</IfModule>
```

另一方面，如果需要一个条件语句，该条件语句恰恰与上面所说的相反，你所需要做的全部事情是在模块名称前面插入一个！（感叹号）。注意，模块名称参数是在编译该模块时使用的文件名（例如，mod_rewrite.c）。<IfModule>段是可以嵌套的；这是一种可以

实现简单的多模块条件测试的方法。

4.2.13 Options

语　　法：Options [+|-]option [+|-]option ...

使用范围：服务器配置，虚拟主机，目录，.htaccess

覆　　盖：Options

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

Options 命令控制在特定的目录中哪些服务器功能是可以得到的。当此命令被设置为 None 时，在此命令的使用范围内禁止所有的额外功能。此命令的所有可能设置是：

- None——没有选项。
- All——除了 MultiViews 之外的所有选项。
- ExecCGI——允许执行 CGI 脚本。
- FollowSymLinks——服务器在目录中跟随符号链接。然而，服务器不改变被用来匹配<Directory>段的路径名。
- Includes——允许使用 SSI 命令。
- IncludesNOEXEC——可以将一组受限制的 SSI 命令嵌入到 SSI 页面中。不允许使用的 SSI 命令包括#exec 和#include。
- Indexes——如果请求映射到目录的 URL，并且在该目录中没有用 DirectoryIndex 指定索引文件（例如，index.html），那么服务器返回格式化的目录列表。
- SymLinksIfOwnerMatch——服务器只跟踪这样一些目标文件或目录的符号链接，这些目标文件或目录的所有者和符号链接的所有者相同。
- MultiViews——允许使用基于文档语言的内容协商。

使用+号和-号表示允许使用或禁止使用在 Options 命令中的选项。让我们看一个例子，使这一问题更清楚些。下面的配置段显示了在单独的配置文件（例如 access.conf）中的两个目录容器：

```
<Directory /www/myclient/public/htdocs>
Options Indexes MultiViews
</Directory>

<Directory /www/myclient/public/htdocs>
Options Includes
</Directory>
```

这样的设置方式只对 /www/myclient/public/htdocs 设置 Includes 选项。然而，如果按如下所示在第二个<Directory>段中使用+号和-号：

```
<Directory /www/myclient/public/htdocs>
Options +Includes -Indexes
</Directory>
```

这就对指定的目录设置了 MultiViews 和 Includes 选项。

请注意，当应用多个 Options 命令时，最窄的使用范围的优先级总是高于较广的使用范围。例如：

```
ServerName
Options  ExecCGI  Includes
<VirtualHost>  11.22.3311.22.33.1>
ServerName  www.myclient.com
Options  -ExecCGI  -Includes
<Directory  /www/myclient/public/htdocs/ssi>
      Options Includes
</Directory>
</VirtualHost>
```

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

在此例子中，主服务器通过将 Options 命令设置为 ExecCGI 和 Includes 而允许执行 CGI 和 SSI。然而，虚拟主机 www.myclient.com 在它自己的 Options 命令中使用了-ExecCGI 和-Includes 设置，从而禁止这两个选项。最后，虚拟主机对于/www/myclient/public/htdocs/ssi 目录使用另一个 Options 命令，这就允许执行 SSI。注意，Includes 是为/www/myclient /public/htdocs/ssi 目录设置的唯一选项。

正如从上面看到的，如果 Options 命令使用了+号或-号，那么，这些值就被添加到当前的 Options 列表中，或被从此列表中删除。而另一方面，如果 Options 命令没有使用相对的+号或-号，那么，该容器的值完全覆盖任何以前的 Options 命令的值。

4.2.14 ResourceConfig

语　　法：ResourceConfig filename

缺 省 值：ResourceConfig conf/srm.conf

使用范围：服务器配置，虚拟主机

此命令与 AccessConfig 命令完全相同。如果想禁止装入缺省的 conf/srm.conf 文件，只需要在你的 httpd.conf 文件中按照如下的方式设置该命令：

```
ResourceConfig  /dev/null
```

4.2.15 ServerAdmin

语　　法：ServerAdmin email-address

使用范围：服务器配置，虚拟主机

此命令指定电子邮件地址，该地址和很多由服务器发出的错误消息一起出现。如果你建立大量的虚拟 Web 站点，那么你需要为每个虚拟主机使用不同的电子邮件地址，因此，你可以立刻知道问题报告程序提供的错误信息来自哪一个服务器。

Tip

为了使你的虚拟站点更为真实，可使用不包含主机部分的虚拟站点地址作为电子邮件地址。例如，如果你的公司是称为 mycompany.net 的 Internet 服务提供商（ISP），而且你有一个称为 www.myclient.com 的客户站点，那么，请将 www.myclient.com 站点的 ServerAdmin 命令设置为 user@myclient.com 这样的地址，例如 webmaster@myclient.com，而不是 webmaster@mycompany.net。这样，当服务器显示错误消息给访问 www.myclient.com 的用户时，访问者将看到属于 myclient.com 的电子邮件地址。这将被认为更为真实。

要找到如何配置邮件服务器使虚拟主机支持 SMTP 邮件服务方面的内容，请参看第 6 章。

4.2.16 ServerName

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

语　　法：ServerName fully qualified domain name

使用范围：服务器配置，虚拟主机

ServerName 命令设置服务器的主机名称。当没有使用此命令时，Apache 在启动时通过发出 DNS 请求来试图确定主机名称。然而，取决于你的 DNS 设置，这可能是不可取的。因为，如果你的服务器有规范的名称记录，Apache 查询的结果可能会为你的服务器选择你不想要的名称。因此，最好是将此命令设置为任何自己喜欢的主机名称。

确保输入完整的域名，而不使用简称。例如，如果你有称为 wormhole.mycompany.com 的主机，不应该将 ServerName 设置为 wormhole。有效的选择是：

```
ServerName  wormhole.mycompany.com
```

4.2.17 ServerRoot

语　　法：ServerRoot directory-filename

缺 省 值：ServerRoot /usr/local/apache

使用范围：服务器配置

ServerRoot 命令设置保存服务器文件的目录。不要将此命令与 DocumentRoot 命令混淆了，后者用于将服务器指向 Web 内容。ServerRoot 目录是用于找到所有的服务器配置文件和日志文件的。标准的发行版包括了在 ServerRoot 指定的目录下面的 conf、src 和 logs 目录。然而，如果没有指定 ServerRoot 命令，则可以使用-d 命令行选项来告诉 Apache 你的 ServerRoot 目录是什么。

注意，AccessConfig 和 ResourceConfig 命令使用 ServerRoot 目录来找到配置文件。例如，如果将这些命令设置为下面的样子：

```
ServerRoot  /usr/local/httpd
AccessConfig   conf/access.conf
ResourceConfig  conf/srm.conf
```

那么，AccessConfig 和 ResourceConfig 命令将告诉 Apache 分别装入/usr/local/httpd/conf/access.conf 和/usr/local/httpd/conf/srm.conf 文件。

4.2.18 DefaultType

语　　法：DefaultType mime-type

缺 省 值：DefaultType text/html

使用范围：服务器配置，虚拟主机，目录，.httaccess

覆　　盖：FileInfo

用此命令建立缺省的内容类型，以便在 Apache 收到所要求的文件类型为未知的文档

请求时（换句话说，不能从供该服务器使用的 MIME 类型映射中确定它的类型），它会使用预先确定的缺省类型。

例如，如果有一个目录，在其中有大量没有扩展名的文本文件，则可以在指向这一目录的<Directory>容器里面使用 DefaultType 命令。在这种情况下，将 DefaultType 设置为 text/plain 使服务器告诉客户端（Web 浏览器端）这些是纯文本文件。下面是一个例子：

```
<Directory  /www/mycompany/public/htdocs/plaindata>
DefaultType  plain/text
</Directory>
```

其中，在/www/mycompany/public/htdocs/plaindata 目录下的所有文件都作为纯文本文件对待。

4.3 性能和资源配置命令

这些命令用来调整 Apache，使它获得更高的性能和得到更好的控制。你可以用很多方法调整 Apache 进程。注意，所有这些命令几乎都要求你对你的系统在操作系统、硬件等方面如何工作有清楚的了解，因此，浏览你的操作系统手册来学习你的系统如何限制进程使用系统资源，以及如何控制 TCP/IP 的连接等是有帮助的。

在这一节中将命令进一步划分为几项子功能。

4.3.1 控制 Apache 进程

下面的命令被用来控制 Apache 如何在系统中执行。使用这些命令还使你能够控制 Apache 如何使用系统中的资源。例如，你可以决定在系统中要运行多少个子服务器进程，或者允许 Apache 在 Windows 平台上使用多少线程。

当配置这些命令时，要记住一些事情：

♦运行的进程越多，CPU 承受的负载越重。

♦运行的进程越多，需要的 RAM 越多。

♦运行的进程越多，使用的操作系统资源（例如，文件访问描述符、共享缓冲区）越多。

当然，更多的进程也表示可以服务更多的请求，因此访问你的站点的机会就越多。因此，设置这些命令需要在经验、需求和可用资源等方面的基础上进行综合考虑。

4.3.1.1 ServerType

语　　法：ServerType inetd | standalone

缺 省 值：ServerType standalone

使用范围：服务器配置

ServerType 命令告诉 Apache 它在 UNIX 系统上应该如何运行。对此命令有两个值：inetd 和 standalone。

使用 inetd 值表示通过称为 inetd 的系统进程来运行服务器。在这样的系统中，对于每

个接收到的 HTTP 连接请求，Apache 服务器启动新的拷贝，在 Apache 服务器服务该连接请求后，它就退出。这增加了系统为服务一个请求必须完成的工作量，因此，它不是高效率的选择。然而，由于 inetd 已经存在很长一段时间，因此对它有各种安全选项（例如 TCP wrapper）可供使用。很多的 Internet 服务器管理员喜欢用这种方法来运行各种服务。对于信息流量低的站点，在性能方面的差异是不明显的。

使用本文内容，请尊重相关知识产权。

Cross-Reference

如果你对设置这种服务器感兴趣，请阅读第 3 章，它详细说明了这一选项。

Note

由 inetd 运行的 Apache 服务器并不会自动地使它比独立的服务器更安全。因为 inetd 已经存在很长一段时间，所以很多经验丰富的管理员都认为它不容易被攻击。

使用 standalone 值将使你在独立模式下运行你的系统，在此模式下，Apache 作为守护进程来运行。换句话说，由主 Apache 服务器侦听到指定端口的连接请求，并在新的连接请求到来的时候，启动子 Apache 进程来服务该请求，而它自己从不服务任何 HTTP 请求。使用在 User 和 Group 命令中指定的用户 ID 和组 ID 运行子进程。查阅这些命令，了解关于它们设置的更多内容。对于繁忙、高信息流量的站点，我建议使用此选项。

4.3.1.2 StartServers

语　　法：`StartServers number`

缺 省 值：`StartServers 5`

使用范围：服务器配置

StartServers 命令只在 Apache 作为独立服务器运行时才有用。换句话说，你需要将 ServerType 设置为 standalone，此命令才生效。

此命令设置在启动时建立的 Apache 服务器子进程数目。注意，在确定时间段内所需的子 Apache 服务器进程数目是被动态控制的。主 Apache 服务器（守护进程）在出现高请求负载时，启动新的子进程。子进程的实际数目由 MinSpareServers、MaxSpareServers 和 MaxClients 命令指定的值所控制。因此，通过调整此参数，对系统性能的影响不明显。

注意，当在 Microsoft Windows 系统上使用时，此命令设置运行的子进程总数。由于 Apache 的 Windows 版是多线程的，一个进程就处理了全部的请求。余下的进程被保持在预备状态，直到主进程结束。

4.3.1.3 ThreadsPerChild

语　　法：`ThreadsPerChild number`

缺 省 值：`ThreadsPerChild 50`

使用范围：服务器配置（Windows）

兼 容 性：只可用在 Apache 1.3 和以后的用于 Windows 的版本中

Apache 的 Windows NT/95 版是多线程服务器。ThreadsPerChild 命令告诉服务器它应该使用多少个线程。它还决定服务器在任意给定时间内能够处理的最大连接数目。因此，

此值应该被设置为适当高的值，以允许可能达到的最大请求数目。

4.3.1.4 SendBufferSize

语　　法：SendBufferSize bytes

使用范围：服务器配置

此命令将 TCP 发送缓冲区的容量设置为指定的字节数。在高性能网络中，将此命令值设置得比操作系统的缺省值更高可以提高服务器的性能。

4.3.1.5 ListenBacklog

语　　法：ListenBacklog backlog

缺 省 值：ListenBacklog 511

使用范围：服务器配置

兼 容 性：只可用在 Apache 1.2.0 之后的版本中。

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

通过设置挂起的连接队列的最大长度，此命令使你能够对已知的名为 Denial of Service（DOS）的安全攻击采取防卫措施。如果你检测到自己正受到 TCP SYN flood（DOS）的攻击，那你可以增加此值；否则，你可以不改变它的值。

4.3.1.6 TimeOut

语　　法：TimeOut number

缺 省 值：TimeOut 300

使用范围：服务器配置

你已经了解，Web 是真正的客户/服务器系统，Apache 服务器在其中响应请求。请求和响应通过数据包进行传输。Apache 必须知道对特定数据包要等待多长的时间。此命令能以秒为单位配置时间。你在这里指定的时间是 Apache 在断开一个连接之前将等待的最长时间。缺省的设置允许 Apache 在将它自己与客户端断开之前等待 300 秒。然而，如果在一个低速网络上，你可能需要提高此超时值，以减少断开连接的次数。

当前，此超时设置适用于：

- 接收 GET 请求所花费的时间总量。
- 在 POST 或 PUT 请求中接收两个 TCP 数据包之间的时间总量。
- 在响应 TCP 数据包传送时两个 ACK 之间的时间总量。

4.3.1.7 MaxClients

语　　法：MaxClients number

缺 省 值：MaxClients 256

使用范围：服务器配置

此命令限制 Apache 能够同时服务的请求的数目。因为 Apache 对每个请求都使用一个子服务器，因此这也是对能够同时存在的子服务器数目的限制。缺省的限制实际上是在 Apache 源发行版中的 httpd.h 文件中设置的硬限制。此设置对大多数从典型到中级负载的站点都能工作得很好。Apache 编程人员将此硬限制放在此处有两个原因：他们不想使服务器填满内核表而使系统崩溃，而且此最大限制使记分板文件保持足够小，能方便地读取。

然而，如果有一个高性能的服务器系统，并有必要的带宽，那你可以通过修改 httpd.h

个 TCP 连接可以用于多个事务处理。在一般情况下，每个 HTTP 请求和响应使用一个独立的连接。这表示服务器每次收到请求时，都打开一个连接并随后关闭它。一旦服务器收到了请求，它就打开另一个 TCP 连接来响应，最后在完成服务后关闭该连接。这种方法降低了服务器的性能。对多个事务处理重复使用单独的连接，减少了重复设置和关闭 TCP 连接所需要的系统开销，因而提高了性能。

然而，要建立持续连接，服务器和客户都需要有相应的连接工具。大多数浏览器，如 Netscape Navigator 和 Microsoft Internet Explorer，都有内置的 KeepAlive 功能。

注意，不是所有的事务处理都能充分利用持续连接。对持续连接的要求之一是传送的资源必须有已知的容量。很多 CGI 脚本、SSI 命令以及其他动态生成的内容在传送之前都没有已知的长度，它们就不能利用此功能。

使用本系列图书
请尊重相关知识产权

4.3.2.1 KeepAlive

语　　法：(Apache 1.2) KeepAlive On | Off

缺 省 值：(Apache 1.2) KeepAlive On

使用范围：服务器配置

兼 容 性：只可用在 Apache 1.1 和以后的版本中

此命令使你能够在 Apache 中打开（或关闭）对 TCP 的持续连接。

Note

旧的 Apache 服务器（1.2 以前的版本）需要使用数字值，而不是 On/Off。该值对应于需要 Apache 对每个请求服务的最大次数。应强行实施限制，以防止一个客户占用服务器的所有资源。在旧的 Apache 版本中使用 0 值禁止 KeepAlive。

4.3.2.2 KeepAliveTimeout

语　　法：KeepAliveTimeout seconds

缺 省 值：KeepAliveTimeout 15

使用范围：服务器配置

如果设置了 KeepAlive 命令，则可以使用此命令限制 Apache 在关闭连接之前等待后继请求的秒数。一旦收到一个请求，就开始对 Timeout 命令指定的超时值计数。

4.3.2.3 MaxKeepAliveRequests

语　　法：MaxKeepAliveRequests number

缺 省 值：MaxKeepAliveRequests 100

使用范围：服务器配置

兼 容 性：只可用在 Apache 1.2 和以后的版本中

MaxKeepAliveRequests 命令限制在 KeepAlive 处于打开状态时允许每个连接服务的请求数。如果它被设置为 0，则允许服务无限多的请求。建议设置较高的值，以优化服务器性能。

4.3.3 控制系统资源

Apache 在允许控制其系统资源（例如 CPU 时间以及内存）消耗量方面是相当灵活的。这些控制特征易于使 Web 服务器系统更为可靠和更具有响应能力。很多典型的黑客攻击试图使 Web 服务器象贪婪者一样消耗全部的系统资源，从而使系统对外界没有反应和几乎停止了工作。Apache 提供一组命令来与这种情况作斗争。下面讨论这些命令。

超星阅览器提醒 您作者相关版权产权！ 请尊重相关版权！

4.3.3.1 RLimitCPU

语　　法：RLimitCPU　n | 'max' [n | 'max']

缺 省 值：未设置；使用操作系统的缺省值

使用范围：服务器配置，虚拟主机

兼 容 性：只可用在 Apache 1.2 和以后的版本中

此命令用来控制 Apache 的 CPU 占用时间。该命令取两个参数。第一个参数为所有的进程设置软的资源限制，而第二个参数设置最大资源限制。提高最大资源限制要求服务器作为 root 运行，或处于系统的初始化阶段。第二个参数是可选的。

每一个参数都有两个可能的值：

- n 是每个进程使用 CPU 的秒数
- max 是操作系统允许使用的最大资源限制

4.3.3.2 RLimitMEM

语　　法：RLimitMEM　n | 'max'　[n | 'max']

缺 省 值：未设置；使用操作系统的缺省值

使用范围：服务器配置，虚拟主机

兼 容 性：只可用在 Apache 1.2 和以后的版本中

此命令限制 Apache 进程的内存（RAM）占用量。该命令取两个参数。第一个参数为所有的进程设置软的资源限制，而第二个参数设置最大资源限制。提高最大资源限制要求服务器由 root 用户启动。第二个参数是可选的。

每一个参数都有两个可能值：

- n 是每个进程使用内存的字节数
- max 是操作系统允许使用的最大资源限制

4.3.3.3 RLimitNPROC

语　　法：RLimitNPROC　n | 'max'　[n | 'max']

缺 省 值：未设置；使用操作系统的缺省值

使用范围：服务器配置，虚拟主机

兼 容 性：只可用在 Apache 1.2 和以后的版本中

此命令设置每个用户并发进程的最大数目。该命令取一个或两个参数。第一个参数为所有的进程设置软的资源限制，而第二个参数设置最大资源限制。每个参数可以取下面两个值中的任一个：

- n 是进程数

♦max 是操作系统允许使用的最大限制

提高最大资源限制要求服务器由 root 运行，或处于系统的初始化阶段。

如果使用同一个用户 ID 运行 CGI 进程与服务器进程，使用此命令将限制服务器可以启动（或“fork”）的进程数。如果限制值设置得太低，你将在服务器错误日志文件中看到类似下面的消息：

```
"Can not fork process"
```

在这种情况下，应该提高限制值或仅仅将它保持为缺省值。

盗用本复制品
请尊重相关知识产权！

4.3.4 使用动态装卸模块的命令

Apache 在启动时装入所有的预编译模块；然而，它还提供了动态装入和卸载模块的功能，该功能在一些情况下可能有用。当使用下面的动态装卸模块的命令时，你可以改变工作模块的列表，而不必重新编译服务器。

4.3.4.1 ClearModuleList

语　　法：ClearModuleList

使用范围：服务器配置

兼 容 性：只可用在 Apache 1.2 和以后的版本中

你可以使用此命令来清除工作模块的列表，并打开动态模块装卸功能。随后，使用 AddModule 命令来添加所要的模块。

4.3.4.2 AddModule

语　　法：AddModule

使用范围：服务器配置

兼 容 性：只可用在 Apache 1.2 和以后的版本中

可以使用此命令装入当前不使用的预编译模块。服务器可以拥有当前不使用的预编译模块。可以用此命令装入这些模块。服务器有预装入的工作模块列表，此列表可以用 ClearModuleList 命令清除。随后，新的模块可以使用 AddModule 命令添加。

4.4 标准的容器命令

在这一节中，将讨论作为基本 Apache 服务器一部分的标准容器。这些容器被广泛用于将一组其他的命令应用到特定的目录、文件或位置。

<VirtualHost>容器将在本章后面单独一节中进行讨论。

不能随便混合使用容器。一般的指导原则是：

♦<Directory>或<Files>容器用来为文件系统对象（例如文件和目录）指定命令。不能在 .htaccess 文件内使用<Directory>容器，因为 .htaccess 文件只适用于包含它的目录。

♦<Location>容器用来匹配 URL 对象。不能在一个.htaccess 文件内使用此命令。

♦当使用规则表达式形式的命令（例如，<DirectoryMatch>）时，遵照一般形式的命令相同的指导原则。只当确保准确地表述规则表达式时，才使用规则表达式形式的容器。

由于 Apache 早期版本的错误，代理控制仍旧用<Directory>容器来完成。而<Location>容器更适合于代理控制。可能在将来的版本中纠正此问题。然而，除了在概念上难于理解外，事实上它不会导致任何损害。

4.4.1 <Directory>

语　法：<Directory directory> ... </Directory>

使用范围：服务器配置，虚拟主机

<Directory>和</Directory>容器用来包容一组适用于指定目录及其子目录的命令。

可以使用任何允许在目录范围内使用的命令。参数可以是完整的路径名。例如：

```
<Directory /www/mycompany/public/hodocs/download>
Options +Indexes
</Directory>
```

其中，目录/www/mycompany/public/hodocs/download 被用作完整的路径名。本例允许在上述目录中使用目录索引。在指定路径时还可以使用通配符。例如：

```
<Directory /www/mycompany/public/htdocs/downloa?>
Options +Indexes
</Directory>
```

其中，? 将匹配任意的单个字符；因此，诸如/www/mycompany/public/hodocs/ download 和/www/mycompany/public/hodocs/downloaD 这样的目录都将被匹配。可以使用*（星号）来匹配除了/（斜杠）字符以外的任意字符序列。还可以使用扩展规则表达式，它们带有附加的~字符。例如：

```
<Directory ~ "^/www/.*/">
```

将匹配在/www/下面的任何子目录。

注意，基于规则表达式的<Directory>容器直到全部正常的（也就是说，不带有规则表达式）<Directory>容器和.htaccess 文件被应用之后才被应用。随后，所有的规则表达式按照它们出现在配置文件中的次序被测试。

有关规则表达式的详细解释，请参见附录 C。

注意，如果为同一个目录空间指定了多个<Directory>容器，范围最小的<Directory>容器将首先被应用。例如：

```
<Directory /www>
AllowOverride None
</Directory>

<Directory ~ "/www/mycompany/public/htdocs/*">
AllowOverride FileInfo
```

```
</Directory>
```

根据此设置，当要求/www/mycompany/public/htdocs/somefile.cvs 的请求到来时，Apache 禁止用于/www 的针对目录的访问控制文件（.htaccess），并允许使用/www/mycompany/public/htdocs 目录的针对目录的访问控制文件。它还接受来自/www/mycompany/ public/htdocs/.htaccess 文件中的任何 FileInfo 命令，例如 DefaultType。

使用本复制品请尊重相关知识产权!

4.4.2 <DirectoryMatch>

语　　法：`<DirectoryMatch regex> ... </DirectoryMatch>`

使用范围：服务器配置，虚拟主机

兼 容 性：只可用在 Apache 1.3 和以后的版本中

除了它使用规则表达式作为参数和不需要~字符以外，此容器与<Directory>容器完全相同。<DirectoryMatch>和</DirectoryMatch>容器用来包容一组只适用于指定目录及其子目录的命令。例如：

```
<DirectoryMatch "^/www/mycompany/public/htdocs/[A-Z] {8}/*">
```

将匹配/www/mycompany/public/htdocs 目录下的所有用 8 个大写字符作为名称的子目录。因此，/www/mycompany/public/htdocs/AAAABBBB/将匹配上述的规则表达式。有关规则表达式的更详细内容，请参见附录 C。

4.4.3 <Files>

语　　法：`<Files filename>...</Files>`

使用范围：服务器配置，虚拟主机，.htaccess

兼 容 性：只可用在 Apache 1.2 和以上的版本中

要通过文件名控制访问，则需要使用此命令。在读取<Directory>段和 .htaccess 文件之后及<Location>段之前，<Files>段按照它们在配置文件中出现的次序进行处理。filename 参数应该包括文件名或通配符字符串；其中？匹配任意单独的字符，而*匹配除了/字符之外的任意字符序列。通过使用~字符，可以使用扩展规则表达式检查此参数。例如：

```
<Files ~ "\.(zip|tar|tgz|arj|zoo)$">
```

将匹配任何带有.zip、.tar、.tgz、.arj 和.zoo 扩展名的文件。与<Directory>段和<Location>段不同，<Files>段可以在.htaccess 文件内使用。当在 .htaccess 文件内使用<Files>段时，不需要添加路径名，因为.htaccess 文件只适用于包含它的目录。

4.4.4 <FilesMatch>

语　　法：`<FilesMatch regex>...</FilesMatch>`

使用范围：服务器配置，虚拟主机，.htaccess

兼 容 性：只可用在 Apache 1.3 和以上的版本中

除了它使用规则表达式作为参数外，此容器与<Files>容器是完全相同。例如：

```
<FilesMatch "\.(zip|tar|tgz|arj|zoo)$">
```

将匹配任何带有.zip、.tar、.tgz、.arj 和.zoo 扩展名的文件。注意，在此命令中使用规则表达式，不需要带~字符。

4.4.5 <Location>

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

语　　法：<Location URL>...</Location>

使用范围：服务器配置，虚拟主机

此命令提供通过 URL 的访问控制。在读取<Directory>和 .htaccess 文件之后，<Location>容器按照它们在配置文件中出现的次序进行处理。

URL 参数不需要 http://servername。它可以使用通配符；例如？（匹配任何单个字符）或*（匹配除了 / 字符以外的任何字符序列）。你还可以在表达式前面加~字符来使用扩展规则表达式。例如：

```
<Location ~ "/(my|your)/file">
```

将匹配诸如/my/file 或/your/file 这样的 URL。

4.4.6 <LocationMatch>

语　　法：<LocationMatch regex>...</LocationMatch>

使用范围：服务器配置，虚拟主机

兼 容 性：Location 只可用在 Apache 1.3 和以后的版本中

除了它的参数（URL）是规则表达式外，此命令与<Location>命令相同。而且，它在表达式前面不要求使用~字符。例如：

```
<LocationMatch "/(my|your)/file">
```

将匹配诸如/my/file 或/your/file 这样的 URL。

4.5 虚拟主机专用命令

这些命令用于创建虚拟主机。在缺省情况下，Apache 只对由 SeverName 命令指定的 Web 站点主机提供服务。然而，通过使用虚拟主机容器命令可以使 Apache 对其他的 Web 站点提供服务。注意，在一般的配置命令部分讨论过的很多命令也适用于虚拟主机。

4.5.1 <VirtualHost>

语　　法：<VirtualHost addr[:port]...>... </VirtualHost>

使用范围：服务器配置

兼 容 性：多地址支持只可用在 Apache 1.2 和以后的版本中

此容器命令指定虚拟主机的配置。在<VirtualHost>和</VirtualHost>之间找到的所有命令只适用于指定的虚拟主机。可以使用所有允许在虚拟主机范围内使用命令。当服务器收到对特定虚拟主机上的文档请求时，它就使用包含在<VirtualHost>中的配置命令。

要指定哪个 IP 地址或 IP 名称将用于特定的虚拟主机，你可以使用下面选项中的任何

一项：

- IP 地址。例如：<VirtualHost 11.22.33.44>...</VirtualHost>
- 带有端口号的 IP 地址。例如：<VirtualHost 11.22.33.44:8080>...</VirtualHost>
- 多个 IP 地址。例如：<VirtualHost 11.22.33.1 11.22.33.2>...</VirtualHost>
- 多个带有端口号的 IP 地址。例如：<VirtualHost 11.22.33.1:8000 11.22.33.2:1000>...</VirtualHost>

虽然可以用 IP 名称替换 IP 地址，但不建议使用这种方法。如果确定地址所必要的 DNS 查找因为某种缘故失败了，服务器可能会混乱，而根本不服务该虚拟站点。

可以指定特殊的名称_default_。在这种情况下，此虚拟主机将匹配没有以显式列出在另一个虚拟主机中的任意 IP 地址。在没有任何包含_default_的虚拟主机以及没有出现匹配项时，将使用在任何 VirtualHost 段以外定义的主服务器配置。

如果未指定端口，它将使用由主服务器最近的 Port 命令指定的端口。还可以指定*来匹配该地址的所有端口。

4.5.2 NameVirtualHost

语　　法：NameVirtualHost addr[:port]

使用范围：服务器配置

兼 容 性：只可用在 Apache 1.3 和以后的版本中

如果计划使用基于名称的虚拟主机，则需要使用此命令。尽管 addr 可以是主机名称，但建议你经常使用 IP 地址。

例如，对于名为 www.mycompany.com 的虚拟主机，它使用 IP 地址 11.22.33.44，则该命令和虚拟主机的定义如下：

```
NameVirtualHost 11.22.33.44
<VirtualHost 11.22.33.44>
ServerName www.mycompany.com
</VirtualHost>
```

如果在多个地址上有多个基于名称的主机，则对每个地址重复该命令。例如：

```
NameVirtualHost 11.22.33.44
# First virtual host that corresponds to the above directive
<VirtualHost 11.22.33.44>
ServerName www.mycompany.com
</VirtualHost>

# Second virtual host that corresponds to the above directive
<VirtualHost 11.22.33.44>
ServerName www.friendscompany.com
</VirtualHost>

# Another NameVirtualHost directive for a new set of name-
based virtual hosts that
```

```
# use a different IP.
NameVirtualHost  11.22.33.55>

<VirtualHost  11.22.33.55>
ServerName  www.myclient.com
</VirtualHost>
<VirtualHost  11.22.33.55>
ServerName  www.herclient.com
</VirtualHost>
```

其中，第一个NameVirtualHost 命令用于www.mycompany.com和www.friendscompany.com虚拟主机。第二个NameVirtualHost 命令用于www.myclient.com和www.herclient.com虚拟主机。可以有选择地指定端口号，在其上使用基于名称的虚拟主机。例如：

```
NameVirtualHost  11.22.33.44:8080
```

4.5.3 ServerAlias

语　　法：ServerAlias host1 host2 ...

使用范围：虚拟主机

兼 容 性：只可用在 Apache 1.1 和以后的版本中。

当有多个 IP 名称（在 DNS 数据库中的 CNAME 记录）的基于名称的虚拟主机时，可以使用单独的虚拟主机命令定义所有的名称。例如：

```
NameVirtualHost  11.22.33.55>

<VirtualHost  11.22.33.55>
ServerName  www.myclient.com
ServerAlias  www.sac-state.edu  www.csu.sacramento.edu
</VirtualHost>
```

在这里，www.sac-state.edu 和 www.csu.sacramento.edu 是 www.csus.edu 的别名。还可以在定义别名时使用通配符，例如*。

4.5.4 ServerPath

语　　法：ServerPath pathname

使用范围：虚拟主机

兼 容 性：只可用在 Apache 1.1 和以后的版本中。

此命令为主机设置遗留的 URL 的路径名称，它和基于名称的虚拟主机一起使用。在典型情况下，它被用来支持非 HTTP/1.1 的浏览器。

有关支持非 HTTP/1.1 浏览器的详细内容，请参看第 6 章。

4.6 记录日志命令

对任何运行 Apache 的系统来说，记录服务器的事务处理是必不可少的工作。服务器日志向你提供了有价值的信息。例如，谁访问过你的 Web 站点，他访问了哪些页面，以及服务器产生了什么错误。

4.6.1 ErrorLog

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

语　　法：ErrorLog filename
缺 省 值：ErrorLog logs/error_log
使用范围：服务器配置，虚拟主机

此命令用来指定记录服务器错误消息的日志文件的名称。如果文件名不以斜杠（/）开始，就认为它是相对于 ServerRoot 的路径。

如果需要禁止记录错误消息，则可以使用下面的命令：

```
ErrorLog /dev/null
```

你的服务器日志目录的权限设置应该只允许 Apache 用户（由 User 命令指定）进行读/写访问，这是非常重要的。允许任何其他的人写此目录可能导致安全漏洞。

4.6.2 ScoreBoardFile

语　　法：ScoreBoardFile filename
缺 省 值：ScoreBoardFile logs/apache_status
使用范围：服务器配置

此命令设置用于存储内部进程数据文件的路径。如果文件名不是以斜杠（/）开始，就认为它是相对于 ServerRoot 的路径。

此文件由主服务器进程用来与子进程通信。如果要找出你的系统是否需要此文件，则只需运行 Apache 服务器，并检查是否在指定的位置创建了一个文件。如果你的系统体系结构需要该文件，必须确保在同一时刻此文件不被多于一个 Apache 调用所使用。此外，确保没有其他用户拥有此文件或保存此文件的目录的读或写访问权限。

因为进程必须完成磁盘 I/O 以进行通信，这可能导致性能瓶颈。因此，如有可能的话，应该为此文件创建 RAM 磁盘。有关细节请参阅操作系统手册。

4.6.3 PidFile

语　　法：PidFile filename
缺 省 值：PidFile logs/httpd.pid
使用范围：服务器配置

通过此命令告诉 Apache 将主服务器（也就是守护进程）的进程 ID 写入一个文件。如果文件名不是以斜杠（/）开始，就认为它是相对于 ServerRoot 的路径。PidFile 只在独立模式下使用。

注意，它的主要用途是便于 Apache 管理员找到主 Apache 服务器的 PID。为了发送信号给服务器，此 PID 是需要的。例如，如果 PID 文件被保存在/usr/local/httpd/logs 目录下，并且它的名称是 httpd.pid，则通过在 shell 提示符后面（作为 root 用户）发出 SIGHUP 信号，管理员可以按如下所示强制 Apache 服务器重新读取它的配置：

```
kill -HUP `cat /usr/local/httpd/logs/httpd.pid`
```

同一个命令可以使 Apache 重新打开 ErrorLog 和 TransferLog。

您好！欢迎使用
请尊重相关知识产权！

和其他任何日志文件一样，要确保 PID 文件不被除服务器进程外的任何人写入，或甚至是读取。为了得到更好的安全保证，应该使日志目录只能由 Apache 服务器用户读取/写入。

4.6.4 LockFile

语　　法：LockFile filename

缺 省 值：LockFile logs/accept.lock

使用范围：服务器配置

如果使用 USE_FCNTL_SERIALIZED_ACCEPT 或 USE_FLOCK_SERIALIZED_ACCEPT 选项编译 Apache，则使用了一个锁文件。可以使用此命令来设置到锁文件的路径。确保只有 Apache 服务器拥有对此文件的读和写访问权限。

将锁文件保存在网络文件系统（NFS）上装配的分区中是不明智的，因为已经知道：当 NFS 涉及文件锁定和安全时是有问题的。

4.7 认证和安全命令

下面讨论的认证和安全命令使你能够为服务器定义访问策略。

4.7.1 AllowOverride

语　　法：AllowOverride override override ...

缺 省 值：AllowOverride All

使用范围：目录

此命令告诉服务器在 .htaccess 文件（由 AccessFileName 指定的）中的哪些命令可以覆盖在配置文件中较早出现的命令。

当 Override 被设置为 None 时，服务器不读取由 AccessFileName 指定的文件（缺省的.htaccess 文件）。这能够加快服务器的响应时间，因为服务器不需要为每个请求寻找由 AccessFileName 指定的文件（详细的内容请参见 AccessFileName 部分）。

然而，如果确实需要基于 AccessFileName 的控制，你可以指定一个或多个选项。这些覆盖选项是：

超星阅览器提醒您 使用本复制品 请尊重相关知识产权！

♦AuthConfig——允许使用认证命令（例如 AuthDBMGroupFile、AuthDBMUserFile、AuthGruopFile、AuthName、AuthType、AuthUserFile 以及 require）

♦FileInfo——允许使用控制文档类型的命令（例如 AddEncoding、AddLanguage、AddType、DefaultType、 ErrorDocument 以及 LanguagePriority）

♦Indexes——允许使用控制目录索引的命令（例如 AddDescription、AddIcon、AddIconByEncoding 、 AddIconByType 、 DefaultIcon 、 DirectoryIndex 、 FancyIndexing 、 HeaderName、IndexIgnore、IndexOptions 以及 ReadmeName）

♦Limit——允许使用控制主机访问的命令（allow、deny 以及 order）

♦Options——允许使用控制特定目录功能的命令（Options 和 XBitHack）

4.7.2 AuthName

语　　法：AuthName label

使用范围：目录，.htaccess

覆　　盖：AuthConfig

此命令为要求认证的资源（例如一个目录）设置标号。当提示输入用户名和口令以允许访问被控制的资源时，通常由 Web 浏览器将此标号显示在弹出式对话窗口中。没有提供缺省的标号。此标号的主要用途是通知客户端的用户，他们试图访问什么资源。例如：

```
AuthName Secured Game Zone
```

将通知用户，他们在请求进入站点的 Secured Game Zone 区域。

注意，为了此命令能够工作，它必须和 AuthType、require 命令以及诸如 AuthUserFile、和 AuthGroupFile 这样的命令一起使用。

4.7.3 AuthType

语　　法：AnthType type

使用范围：目录，.htaccess

覆　　盖：AuthConfig

此命令为目录选择用户认证类型。在 Apache 中当前只实现了基本的 HTTP 认证类型。此类认证还不能满足重要场合的需要：口令和用户名称是以纯文本（明文）方式被传送。将为每个访问同一受限目录或其子目录的后继请求重新传送口令和用户名。

为了能够正常工作，此命令必须和 AuthName 一起使用，并要求其他的命令，例如 AuthUserFile 和 AuthGroupFile。

4.7.4 require

语　　法：require entiry-name entity entity...

使用范围：目录，.htaccess

覆　盖：AuthConfig

通过使用此命令，Apache 可以确定哪些用户或哪些组能够访问受限的目录。可用三类实体名称：

- 用户
- 组
- 合法用户

例如：

```
require user joe jenny
```

它告诉 Apache 只允许 joe 或 jenny 在经过成功认证后进入该区域。只有指定的用户可以访问该目录。基于组访问要求的例子如下：

```
require group my-group your-group his-group her-group
```

只有在指定组中的用户才能够访问该目录。

```
require valid-user
```

使用上述命令行，所有合法的用户都能够访问该目录。

如果 require 命令出现在<Limit>段中，它将限制对指定方法的访问；否则，它就限制对所有方法的访问。例如：

```
AuthType Basic
AuthName Game Zone Drop Box
AuthUserFile /www/netgames/users
AuthGroupFile /web/ntgames/groups
<Limit GET>
require group coders
</Limit>
```

如果在一个目录的 .htaccess 文件中出现了上述的配置，则只有名为 coders 的组才被允许通过 HTTP GET 方法来访问该目录以获得文件。为了能够正确工作，require 命令必须和 AuthName 和 AuthType 命令以及诸如 AuthUserFile 和 AuthGroupFile 这样的命令一起使用。

有关认证的详细内容，请参见第 10 章。

4.7.5 Satisfy

语　法：Satisfy 'any' | 'all'

缺 省 值：Satisfy all

使用范围：目录，.htaccess

兼 容 性：只可用在 Apache 1.2 和以后的版本中

如果已经建立基本的 HTTP 认证配置，而在其中同时使用了 allow 和 require 命令，则可以使用此命令告诉 Apache 什么条件才能满足认证的要求。Satisfy 命令的值可以是 all 或 any。如果其值是 all，那么只有在 allow 和 require 都成功的情况下认证才成功。如果其

值是 any，那么，如果 allow 成功了，或者 require 成功了，则认证都成功。

此命令只当访问的特定区域同时受用户名称/口令和客户端主机地址的限制时才是有用的。在这种情况下，缺省的条件（all）要求客户端既要通过地址访问限制，又要输入合法的用户名和口令。在使用 any 选项的情况下，如果客户通过了主机限制或者输入了合法的用户名和口令，则允许其进行访问。此命令可以用来限制一个使用口令区域：与此相应，来自特定的 IP 地址组合（也就是说，一组 IP 地址）的所有客户都不需要输入口令就被允许进行访问。

4.7.6 IdentityCheck

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

语　　法：IdentityCheck boolean
缺 省 值：IdentityCheck off
使用范围：服务器配置，虚拟主机，目录，.htaccess

此命令告诉 Apache 通过与远程用户的 identd（identification 守护进程）进程或类似的进程并遵照 RFC1413 进行对话以记录远程用户的名称。它不是什么有用的命令，因为它不能在所有系统上工作。大多数系统都不运行 identd 进程来向远程服务器提供用户标识。

如果你决定在配置中使用此命令，一定要注意你记录的信息除了用于跟踪之外，在任何其他情况下都不能被信赖。此命令还能导致严重的性能问题，因为服务器不得不为每个请求执行检查。还有，当远程用户没有提供 identd 服务，或者位于防火墙或代理服务器后面的时候，检查进程就会超时。

4.7.7 HostNameLookups

语　　法：HostNameLookups on | off | double
缺 省 值：HostNameLookups off
使用范围：服务器配置，虚拟主机，目录，.htaccess
兼 容 性：double 只可用在 Apache 1.3 和以上的版本中

此命令指示 Apache 允许或禁止为每个请求而进行的 DNS 查找。当被允许时，Apache 将客户的主机名称存储在它运行的每个 CGI 和 SSI 进程的 REMOTE_HOST 环境变量中。

on 和 off 值的含义应该是显而易见的。Double 指定进行双反向查找——也就是说，在执行一个反向查找之后，随后对前面的结果执行前向查找。在前向查找中至少有一个 IP 地址必须匹配原来的地址。然而，CGI 和 SSI 进程都不会从双 DNS 查找中得到结果。

Note

不管将此命令设置为什么值，当 mod_access（将在第 13 章讨论的一个模块）通过主机名而被用于控制访问时，将执行一个双反向查找。

建议保持此命令的缺省设置。这将使 Internet 去掉大量不必要的 DNS 信息流量。如果要将它打开以使日志文件包含 IP 名称，而不是 IP 地址，那么可能需要考虑另一个选项。例如，运行 logresolve 实用程序将 IP 地址解析为 IP 名称。有关更多的细节，请参见第 11 章。

4.7.8 <Limit>

语　　法：<Limit method method ...>... </Limit>

使用范围：任意

此容器被用来封装一组访问控制命令，而这些命令将只适用于指定的 HTTP 方法。列出的方法名称可以是下述方法中的一个或多个：GET、POST、PUT、DELETE 以及 OPTIONS。如果 GET 被使用，它还将限制 HEAD 请求。如果希望限制所有的方法，根本不要在<Limit>命令中包括任何方法。

注意，此容器不能被嵌套，而且<Directory>容器也不能出现在它的里面。

超星阅览器提示
使用本复制品
请尊重相关知识产权！

4.8 小　　结

在本章中，讨论了内核命令和它们的使用范围。这些内核命令在管理高度可配置的 Apache Web 服务器方面提供了大量的功能和灵活性。

在下一章中，将讨论标准的 Apache 模块。

第 5 章　Apache 模块

本章内容:

- 介绍 Apache 模块
- 如何使用各种 Apache 模块

超星阅览器提醒您:
使用本复制品
请尊重相关知识产权!

在前一章中，讨论了内核的 Apache 命令。Apache 还在标准源代码发行版中提供了许多模块。这些模块通过使用各自的命令而提供了大量功能。

在本章中，将讨论这些模块以及它们的命令。为便于阅读，按照字母顺序列出了各个模块。

5.1　mod_access

在缺省情况下，此模块被编译在 Apache 中。它提供基于主机的访问控制命令，这些命令将在第 10 章中讨论。

有关 mod_access 命令的详细内容，请参见第 10 章的“基于主机的访问控制”一节。

5.2　mod_actions

在缺省情况下，此模块被编译。它使你能够运行基于 MIME 类型的 CGI 脚本或 HTTP 请求方法。它提供了下面的命令。

5.2.1　Action

语　　法: `Action MIME-type cgi-script`

使用范围: 服务器配置，虚拟主机，目录，针对目录的配置文件（`.htaccess`）

覆　　盖: `FileInfo`

状　　态: 基本（`Base`）

此命令使你能够对特定的 MIME-type 指定一个操作。此操作通常是处理被请求文件的 CGI 脚本。例如:

```
Action text/html /cgi-bin/somescript.pl
```

这将使 Apache 在 HTML 文件被请求时都要运行指定的脚本。此脚本通过标准的 CGI PATH_INFO 和 PATH_TRANSLATED 环境变量接收 URL 以及被请求文档的文件路径。在开发过滤脚本时，这可能有用。本节将讨论这样的过滤脚本。

当通过 Web 请求文本文件（.txt）时，它将以不太令人满意的格式显示在 Web 浏览器中。这是因为 Web 浏览器不对行间隔作任何方式的翻译。在通常情况下，大多数的文本文件是作为大段出现的。通过使用 Action 命令，你可以开发更好的解决方案。为了使此范例更有趣些，假定你想要开发一种解决方案，该方案不仅能够更好地在 Web 浏览器上显示文本文件，而且还能够在每个文本文件的末尾插入一条版权信息。

要做到这一点，你需要做两件事情。首先，将下面的命令添加到一个 Apache 配置文件中（如 srm.conf）：

```
Action plain/text /cgi-bin/textfilter.pl
```

超星阅览器提醒您：
此电子书受版权保护，请勿复制或传播，尊重知识产权！

然后，再开发作为过滤程序的 Perl 脚本，该过滤程序将以你希望的方式显示文本文件。列表 5-1 显示了这样的脚本。

列表 5-1：textfilter.pl

```
# !/usr/local/bin/perl
#
# Script:textfilter.pl
#
# Purpose:this filter script converts plain text files
#  into an HTML document but keeps the text layout
#  as-is.
#
# $Author$
# $Revision$
# $Id$
# $Status
#

# The copyright message file will be always stored on
# the server's document root
# directory and is called copyright.html.
#
my $copyright_file = $ENV{DOCUMENT_ROOT} . "/copyright.html";

# Get the requested document's path
my $path_translated = $ENV{PATH_TRANSLATED};

# Other variables needed for storing data
my $line;
my @text;
my @html;

# Store the path info and the file name of requested doc in an
array
@filename = split(/\//, $path_translated);
```

```
# Since HTML tags will be used to display the text file,
# lets print the text/html content header.
print "Content-type: text/html\n\n";

# Read the document requested and store the data
# in @text array variable
@text = &readFile($path_translated);

# Now print the following HTML document tags
# these tags will be sent before the actual document content
#
print <<HEAD;
<HTML>
<HEAD>
<TITLE>$filename[-1]</TITLE>
</HEAD>
<BODY BGCOLOR = "white">

<BLOCKQUOTE>

<PRE>

<FONT FACE="Arial">

HEAD

# Now print each line stored in the @text array
# (i.e. the content of the document requested)
#
foreach $line (@text){print $line;}

# Now read the copyright file and store the content
# in the @html array variable
#
@html = &readFile($copyright_file);

# Print each line stored in the @html array (i.e.
# the content of the copyright message file)
#
foreach $line (@html){print $line;}

# Exit the  filter
exit 0;
sub readFile{
#
# Subroutine:readFile
```

超星阅览器提醒您:
使用本图书馆
请尊重相关知识产权!

```
# Purpose: reads a file if it exists or else prints
# an error messages and exists script
#

# Get the name of the passed file name and store
# it in variable $file
my $file = shift;

# Local buffer variable
my @buffer;

# If the file exists then open it and read all the
# lines into the @buffer array variable

if(-e $file){

  open(FP, $file) || die "Can not open $file.";

  while(<FP>){
    push(@buffer, $_);
    }

  close(FP);
  }

 else{
  push(@buffer, "$file is missing.");
}

# Return the content of the buffer.
return (@buffer);
}
```

您的浏览器提醒您:
使用本复制品
请尊重相关知识产权!

此脚本读取被请求的文本文件，并打印出一些在 HTML 标记之间的内容，而这些标记使内容能够按照它们原有的样子显示出来。这一技巧是通过 HTML 标记<PRE>来实现的。在内容被打印出来之后，一个版权消息文件内容被插入到输出的末尾。这就使得版权消息可以和每个被请求的文本文件一起打印出来。图 5-1 显示了样本输出，其中一个文本文件被显示在 Web 浏览器上。

正如你看到的，被请求的文件名作为标题出现。此文档被作为块来引用，并且定制的版权信息也被打印出来。版权信息文件被存储在文档根目录下。在此例子中使用的版权信息是：

```
BLOCKQUOTE>
<CENTER>
<HR>
```

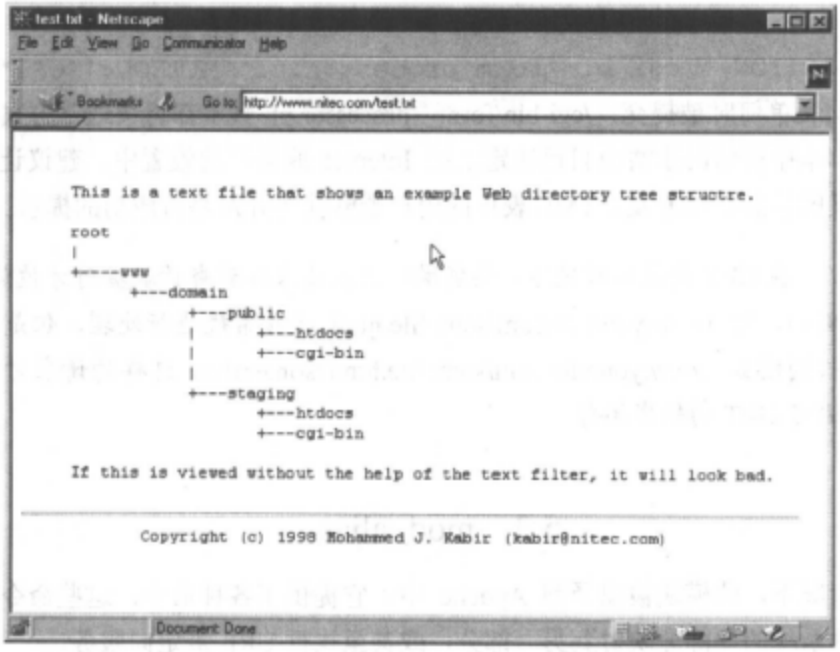

图 5-1　textfilter.pl 的输出

```
Copyritht ÷ 1998 Mohammed J. Kabir ( ) ({Hyperlink
Mail to:kabir@nitec.com})
</CENTER>
</BODY>
</HTML>
```

5.2.2　Script

语　　法：Script method cgi-script

使用范围：服务器配置，虚拟主机，目录

此命令与 Action 命令相似，但它不是将操作和 MIME 类型联系起来，而是将操作和 HTTP 请求方法联系起来，如 GET、POST、PUT 或 DELETE。CGI 脚本通过使用标准的 CGI PATH_INFO 和 PATH_TRANSLATED 环境变量来接收 URL 以及被请求文档的文件路径。此命令定义了缺省的操作。换句话说，如果已经在 Apache 配置文件中（如 srm.conf）定义下面的语句：

```
Script POST /cgi-bin/default_post.pl
```

当通过 HTTP POST 方法提出请求时，除了需要使用由命令指定的缺省操作外，它将按照通常的方式进行处理。例如，下面的 HTML 表单没有将 CGI 脚本指定为它的操作：

```
<FORM METHOD="POST">
Enter Name: <INPUT TYPE=TEXT NAME="name" SIZE=25>
<INPUT TYPE=SUBMIT VALUE="Click Here">
</FORM>
```

如果用户通过此表单提交一个名称，而没有指定 CGI 的脚本来处理信息，在这种情况下，缺省的 POST 操作/cgi-bin/default_post.pl 脚本被运行。然而，如果将<FORM...>标

记变为：

```
<FORM ACTION="/cgi-bin/form_processor.pl" METHOD="POST">
```

则无论此表单何时被提交，/cgi-bin/form_ processor.pl 脚本都将按通常方式被调用。你在缺省的脚本中所做的事情由自己决定。在 Internet 服务商的设置中，建议让缺省的脚本打印有意义的信息，使开发 HTML 表单的用户能够获得开发是否成功的提示。

Note

在 GET 请求的情况下，缺省操作只在请求伴随查询数据时才被使用。例如，对 www.yoursite.com/somefile.html 将按常规进行处理，但是，当收到诸如 www.yoursite.com/somefile.html?some=data 这样的请求时，将运行 GET 的缺省操作。

5.3 mod_alias

在缺省情况下，此模块被编译到 Apache 中。它提供了各种命令，这些命令可以用来将服务器文件系统的一部分映射到另一部分，或甚至执行 URL 重定向服务。

5.3.1 Alias

语　法：Alias URL-path path

使用范围：服务器配置，虚拟主机

此命令使你能够将一个路径映射到文件系统中的任何位置。例如：

```
Alias /data /web/data
```

这将/data 映射到/web/data。因此，当收到诸如 www.yoursite.com/data/datafile.cvs 这样的请求时，将返回名为/web/data/datafile.cvs 的文件。别名路径不一定要在你的文档根树中，因此，在创建别名时，一定注意——你可能会不小心将文件系统的某一部分展示在整个世界面前。

注意，如果在定义别名时使用了尾部 /（斜杠），则访问经过别名处理的目录也需要包含尾部 /。例如：

```
Alias /data/ /web/data
```

这只用来对 www.yoursite.com/data/提供服务，而不对 www.yoursite.com/data 提供服务。在后一种情况下，Apache 寻找一个在文档树下面的 data 目录，而不是/web/data 目录。

5.3.2 AliasMatch

语　法：AliasMatch regex path

使用范围：服务器配置，虚拟主机

它与 Alias 命令相似，但可以使用规则表达式。例如：

```
AliasMatch ^/data(.*) /web/data$1
```

此语句将 www.yoursite.com/data/index.html 匹配/web/data/index.html 文件。

5.3.3 Redirect

语　　法：`Redirect [status] old_URL new-URL`

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

此命令将 URL 请求重定向到另一个 URL。例如：

```
Redirect /data www.your-new-site.com/data
```

此语句将所有包含/data 路径的请求重定向到新 URL。因此，对 www.yoursite.com/data/somefile.txt 的请求将被重定向到 www.your-new-site.com/data/somefile.txt。

仅供阅览参考使用 使用本资料侵犯版权！

Redirect 命令比 Alias 和 ScriptAlias 命令有更高的优先级。在缺省情况下，发送给客户的状态码是 Temp（HTTP 状态码 302）。如果需要指定不同的状态码，则请使用下面的状态码：

- Permanent——告诉客户此重定向是永久的。HTTP 状态码 301 被返回。
- Temp——返回临时的重定向状态码（302）。这是缺省的设置。
- See other——返回 See Other 状态码（303），表明资源已经被替换了。
- Gone——返回 Gone 状态码（410），表明资源已经被永久移走了。当使用此状态码时，应该忽略 URL 参数。

还能够以数字方式提供合法的 HTTP 状态码。如果所提供的状态码在 300 和 399 之间，则新的 URL 必须存在；否则，它必须被忽略。你可能对使用不同的状态码感到奇怪。在将来，客户端可能会足够高明，并以一种更有意义的方式识别出状态码。例如，如果代理服务器收到了一个永久重定向状态码，它能够将此信息存储在缓存中，以便在以后的请求中直接访问这一新的资源。

5.3.4 RedirectMatch

语　　法：`RedirectMatch [status] regex URL`

使用范围：服务器配置，虚拟主机

此命令与 Redirect 命令相似，但它接受规则表达式，而不是简单的旧式 URL。例如：

```
RedirectMatch (.*)\.htm$ www.yourserver.com$1.html
```

此语句将所有以.htm 结尾的请求重定向到相同的以.html 结尾的请求。例如，下面的请求：

```
www.yoursite.com/some/old/dos/files/index.htm
```

被重定向到：

```
www.yoursite.com/some/old/dos/files/index.html
```

5.3.5 RedirectTemp

语　　法：`RedirectTemp old-URL new-URL`

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

此命令与 Redirect 命令相似。它让客户端知道重定向只是临时的。注意，在缺省情况

下，Redirect 命令还生成临时的状态码。

5.3.6 RedirectPermanent

语　　法：RedirectPermanent old-URL new-URL

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

此命令与 Redirect 命令相似。它让客户端知道重定向是永久的。注意，Redirect 命令在缺省情况下生成临时的状态码，但是你可以使用状态码 301 或关键字 permanent，因为此状态码作用与此命令相同。

使用本文内容时请尊重相关知识产权

5.3.7 ScriptAlias

语　　法：ScriptAlias URL-path path

使用范围：服务器配置，虚拟主机

此命令为路径创建一个别名（URL-path ）。还有，在请求中提供的任何文件名都被作为 CGI 脚本来看待，而且服务器会尝试运行该脚本。例如：

```
ScriptAlias /cgi-bin/ /www/nitec/public/cgi-bin
```

这可以用来处理诸如 www.nitec.com/cgi-bin/somescript.pl 这样的请求。如果正确的权限得到验证，则服务器会尝试运行 somescript.pl。注意，ScriptAlias 目录是不能浏览的。

5.3.8 ScriptAliasMatch

语　　法：ScriptAliasMatch regex directory-filename

使用范围：服务器配置，虚拟主机

状　　态：Base

除了使用规则表达式之外，此命令与 ScriptAlias 命令等价。例如：

```
ScriptAliasMatch ^/cgi-bin(.*) /www/nitec/public/cgi-bin$1
```

将与下面的语句做完全相同的事情：

```
ScriptAliasMatch /cgi-bin/ /www/nitec/public/cgi-bin
```

5.4 mod_asis

在缺省情况下，此模块被编译到 Apache 中。它使你能够按照原样发送文档——换句话说，文档在没有 HTTP 头标的情况下被发送到客户端。在没有任何脚本帮助的情况下，这一功能在重定向客户时是有用的。要按照文件原来的样子发送它，需要确保 Apache 配置文件之一（例如，srm.conf）包含了诸如下面的条目：

```
AddType httpd/send-as-is asis
```

此语句将 MIME-type httpd/send-as-is 指定给文件扩展名.asis。如果你创建了名为 foobar.asis 的文件，而且一个客户请求它，则此文件将在不带任何 HTTP 头标的情况下被发送到客户。将合适的头标包括在文件中将是你的工作。例如，如果需要通过.asis 文件提

供重定向机制，你可以创建带有诸如下面头标的文件：

```
Status: 301 Text Message
Location: new-URL
Content-type: text/html
```

例如，列表 5-2 表示称为 redirect.asis 的文件，它将客户重新定向到新的位置。

列表 5-2　redirect.asis

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

```
Status: 301 We have moved.
Location: http://www.our-new-site/
Content-type: text/html

<H1>Notice to Visitors</H1>
Please update your bookmark to point to <A HREF="www.our-new-
site/">www.our-new-site/ </A><BR>
<BR>
Thanks.
```

当客户请求此文件时，301 状态消息告诉客户使用地址信息来重定向请求。不必添加 Date:和 Server:头标，因为服务器会自动添加它们。然而，服务器不会提供最后修改的头标。

5.5　mod_auth

请参见第 10 章。

5.6　mod_auth_anon

请参见第 10 章。

5.7　mod_auth_db

请参见第 10 章。

5.8　mod_auth_dbm

请参见第 10 章。

5.9　auth_external

请参见第 10 章。

5.10 mod_autoindex

此模块在缺省情况下被编译到 Apache 中，当 Apache 收到一个目录请求时，它就查阅由 DirectoryIndex 命令指定的一个或多个目录索引文件。在典型情况下，此文件是 index.html 或 index.htm。然而，在缺少这样的索引文件的时候，Apache 能够生成动态目录列表。此模块使你能够控制 Apache 如何创建动态目录列表。

Apache 生成两类动态目录索引。图 5-2 显示了用于 www.nitec.com/ski 目录的动态目录索引。

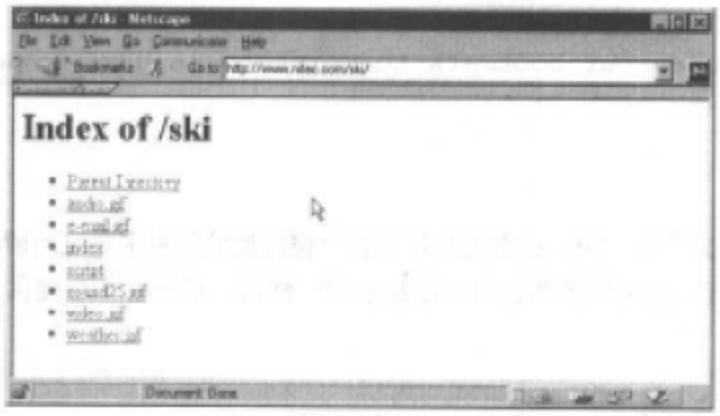

图 5-2 由 Apache 生成的简单目录索引

注意，这是非常简单的索引。现在，看一下图 5-3，它显示相同目录的非常好的索引。通过在此模块中找到的 FancyIndexing 或 IndexOptions 命令，可以得到高级索引。

在此模块中可以得到高级索引和很多其他的索引选项。下面列举用于 mod_authoindex 的命令。

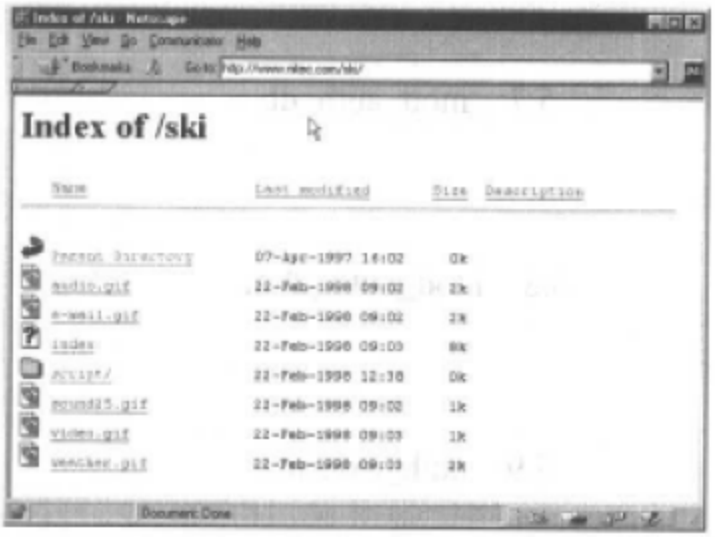

图 5-3 由 Apache 生成的高级目录索引

当 FancyIndexing 处于打开状态时，此命令为带有不完整文件名或通配符文件名的文件设置描述正文。例如：

```
AddDescription "Graphics File" *.gif *.jpeg *.jpg *.bmp
```

这将为所有的 GIF、JPEG、JPG 以及 BMP 文件显示描述正文“Graphics File”。

5.10.5 AddIcon

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

语　　法：AddIcon icon name name...

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

此命令使你能够将图标指定给用于 FancyIndexing 显示的文件和目录名称。例如：

```
AddIcon /icons/picture.gif .gif .jpg .bmp
```

此语句告诉 Apache 在具有诸如.gif、.jpg 以及.bmp 这样扩展名的文件的旁边显示/icons/picture.gif。如果还需要为列出的文件扩展名提供代用正文，可以使用下面的格式：

```
AddIcon (IMG, /icons/picture.gif) .gif .jpg .bmp
```

在这里，IMG 是为非图形界面浏览器显示的代用正文。如果为目录提供图标，可以使用如下的命令：

```
AddIcon /path/to/your/directory/icon ^^DIRECTORY^^
```

与此相似，如果需要为由高级索引方案显示的每个空白行显示一个图标，可以使用：

```
AddIcon /path/to/your/blank/line/icon ^^BLANKICON^^
```

5.10.6 AddIconByEncoding

语　　法：AddIconByEncoding icon mime-encoding mime-encoding...

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

此命令允许将图标指定给 MIME-encoding。例如：

```
AddIconByEncoding /icons/zip.gif x-gzip
```

5.10.7 AddIconByType

语　　法：AddIconByType icon MIME-type MIME-type...

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

此命令也允许将图标指定给 MIME-types。例如：

```
AddIconByType (HTML, /icons/html.gif) text/html
```

5.10.8 DefaultIcon

语　　法：DefaultIcon　url

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

当 AddIcon、AddIconByEncoding 或 AddIconByType 指定的图标都不与一个文件匹配时，将显示缺省的图标。此命令使你能够设置缺省图标。例如：

```
DefaultIcon  /icon/idontknow.gif
```

当图标与文件的联系为未知时，将显示缺省图标 idontknow.gif 。

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

5.10.9 FancyIndexing

语　　法：FancyIndexing　on | off

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

此命令允许或禁止目录的高级索引。可以用 IndexOptions 命令来取得相同的效果。

5.10.10 HeaderName

语　　法：HeaderName　filename

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

如果使用 FancyIndexing，可以在索引列表的前面插入一个文件的内容。此命令允许为这样的插入操作指定文件名称。例如：

```
HeaderName  welcome
```

此语句使得 Apache 在列出的目录中寻找名为 welcome 或 welcome.html 的文件。如果找到了这样的文件，它的内容就被插在实际列表的前面。

5.10.11 IndexIgnore

语　　法：IndexIgnore　file　file...

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

如果需要使一些文件或文件扩展名不出现在目录列表中，可以使用此命令来达到目的。例如：

```
IndexIgnore welcome welcome.html per-directory configuration
(.htaccess)
```

这就确保 Apache 不在目录列表中列出 welcome、welcome.html 或针对目录的配置文件（.htaccess）。.（点）字符自动出现 IndexIgnore 列表中；因此，以“点”字符开头的文件都不列出。然而，我还是喜欢在列表中添加针对目录的配置文件（.htaccess），只是为了感觉更安全。

5.10.12 IndexOptions

语　　法：IndexOptions option option...

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

IndexOptions 命令指定目录索引的行为。它的选项有：

FancyIndexing：它打开目录的高级索引。注意，FancyIndexing 和 IndexOptions 命令将彼此覆盖。

使用本复制品请尊重相关知识产权！

IconHeight[=pixels]：此选项使 Apache 能够在图标的 IMG 标记中包括 HEIGHT=pixels 属性，这使得大多数浏览器更快地加载图标。如果你没有指定像元计数，则使用标准的缺省值。

IconsAreLinks：此选项用于高级索引，用来制作文件名称锚（anchor）的图标部分。

IconWidth[=pixels]：此选项使 Apache 能在图标的 IMG 标记中包括 WIDTH=pixels 属性，这将使大多数浏览器更快地加载图标。如果没有指定像元计数，则使用标准的缺省值

ScanHTMLTitles：如果需要 Apache 为高级索引读取 HTML 文档标题（由<TITLE>和</TITLE>表示），则请使用此选项。然而，如果已经用 AddDescription 命令指定了一个描述，则此选项就不被使用。注意，读取每个文件的内容并搜索标题信息是一项耗费时间的任务，它可能减缓提交目录列表的速度。不推荐使用此选项。

SuppressColumnSorting：在缺省情况下，Apache 为高级目录索引制作可以单击的栏目标题，这将使用户能对该栏中的信息进行排序。而此选项禁止此功能。

SuppressDescription：如果不想在高级目录列表中显示文件描述，请使用此选项。

SuppressHTMLPreamble：如果目录实际包含了由 HeaderName 命令指定的文件，则这一模块通常在标准的 HTML preamble（如<HTML>、<HEAD>等等）后面包括该文件的内容。而 SuppressHTMLPreamble 命令将禁止此行为。

SuppressLastModified：此选项禁止在高级索引列表中显示上次修改的日期。

SuppressSize：此选项禁止在高级索引列表中显示文件的容量。

5.10.13 ReadmeName

语　　法：ReadmeName filename

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

如果需要在高级索引列表的最后插入一个文件，则使用此命令。例如：

```
ReadmeName readme
```

Apache 寻找名为 readme.html 或 readme 的文件，将它插到列表的最后。

5.11　mod_cern_meta

此模块在缺省情况下不被编译。它提供对元信息的支持。元信息或者是附加的 HTTP 头标，例如：

超星阅览器提醒您：
使用本复制品
需尊重相关知识产权！

```
Expires:  Saturday, 3-Oct-98  12:00:00  GMT
```

或者是任意其他的信息，例如：

```
Foo=Bar
```

元信息被存储在一个文件中，并和 HTTP 应答头标一起出现。

5.11.1　MetaFiles

语　　法：MetaFiles　on/off
缺 省 值：MetaFiles　off
使用范围：针对目录的配置文件（.htaccess）
此命令允许或禁止元头标文件处理。

5.11.2　MetaDir

语　　法：MetaDir　directory name
缺 省 值：MetaDir　.web
使用范围：针对目录的配置文件（.htaccess）
此命令用来指定存储元头标文件的目录的名称。例如，如果你有名为/www/mycompany/public/htdocs 的目录，并且需要为该目录存储元信息文件，那么，如果使用 MetaDir 命令的缺省值，则需要创建名为.web 的子目录。.web 目录被用来存储元头标文件。

5.11.3　MetaSuffix

语　　法：MetaSuffix　suffix
缺 省 值：MetaSuffix　.meta
使用范围：针对目录的配置文件（.htaccess）
此命令为元信息文件指定文件的扩展名。例如，如果有名为 mypage.html 的 HTML 文件，则你需要创建 mypage.html.meta（使用此命令的缺省值）来存储你的元头标。mypage.html.meta 必须放在由 MetaDir 命令所指定的目录中。

要使 Apache 能够为名为/www/mycompany/public/htdocs 的目录发送元信息，你需要做下面的一些事情：

1. 在用于/www/mycompany/public/htdocs 目录的 .htaccess 文件中将 MetaFiles 命令设置为打开。还可以在此文件中设置 MetaDir 和 MetaSuffix 命令。
2. 创建名为 .web（假定使用 MetaDir 命令的缺省值）的子目录。
3. 创建带有扩展名 .meta（假定使用 MetaSuffix 命令的缺省值）的文本文件

4. 在此文件中放置你要提供的所有 HTTP 头标。

例如，要为名为/www/mycompany/public/htdocs/mypage.html 的文件提供元头标，需要创建名为/www/mycompany/public/htdocs/.web/mypage.html.meta 的文件。

此文件可以包括如下所示的行：

```
Expires: Saturday, 3-Oct-98  12:00:00  GMT
Anything=Whatever
```

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

5.12 mod_cgi

请参见第 8 章。

5.13 mod_digest

请参见第 10 章。

5.14 mod_dir

此模块在缺省情况下被编译到 Apache 中。通过使用此模块，Apache 能够重定向任何对不包括尾部斜杠字符命令的请求。例如，此模块可以将 www.yoursite.com/somedirectory 重定向到 www.yoursite.com/somedirectory/。它还提供了名为 DirectoryIndex 的命令来帮助建立目录内容的索引。

5.14.1 DirectoryIndex

语　　法：DirectoryIndex local-URL local-URL...

缺 省 值：DirectoryIndex index.html

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

此命令指定 Apache 在创建动态目录索引之前应该搜寻的文件名称。这些文件可以是 HTML 文件也可以是 CGI 脚本等等。缺省的设置使 Apache 能为任何以目录名结尾的请求查找 index.html 文件。例如，对 www.yoursite.com/some/directory/的请求将导致 Apache 去查找名为/some/directory/index.html 的文件。如果该文件存在，则将它的内容提交给客户。如果此文件不存在，Apache 就创建动态的目录列表。

可以指定一个或多个文件来作为缺省的目录索引文件。例如：

```
DirectoryIndex index.html index.htm welcome.html welcome.htm
```

这就告诉 Apache 为每个目录请求查找所有指定的文件。注意，Apache 将以这些文件出现在上述配置中的相同次序（从左到右）来查找它们。换句话说，如果 Apache 找到了 index.html，它将不再寻找 index.htm、welcome.html 和 welcome.htm。你还可以指定 CGI

脚本名称来作为缺省的索引。例如：

```
DirectoryIndex  /cgi-bin/show_index.cgi
```

这将使 Apache 在每次收到目录请求时都运行/cgi-bin/show_index.cgi 脚本。

5.15 mod_env

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

此模块在缺省情况下不被编译。它使你能够将环境变量传递给 CGI 或 SSI 脚本。它提供以下的命令。

5.15.1 PassEnv

语　法：PassEnv variable variable...

使用范围：服务器配置，虚拟主机

此命令告诉模块从服务器自己的环境中传递一个或多个环境变量到 CGI 和 SSI 脚本。例如：

```
PassEnv HOSTTYPE PATH
```

5.15.2 SetEnv

语　法：SetEnv variable value...

使用范围：服务器配置，虚拟主机

此命令设置环境变量，然后将此变量传递给 CGI/SSI 脚本。例如：

```
SetEnv CAPITAL_CITY SACRAMENTO
```

5.15.3 UnsetEnv

语　法：UnsetEnv variable variable...

使用范围：服务器配置，虚拟主机

此命令从传递给 CGI/SSI 脚本的环境变量中删除一个或多个变量。例如：

```
UnsetEnv PATH
```

5.16 mod_expires

在缺省情况下，此模块不被编译。它让你确定 Apache 在服务器响应请求时如何处理 Expires HTTP 头标。Expires HTTP 头标提供了在所请求的资源变为无效时告诉客户端时间的方法。当文档被客户端缓存并需要被再次请求时，这是有用的。大多数高明的客户端将通过检查由 Expires HTTP 头标提供的缓存文档过期时间来确定再次请求文档的有效性。此模块能够控制 Expires HTTP 头标的设置。

5.16.1 ExpiresActive

语　　法：ExpiresActive on | off

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

此命令允许或禁止 Expires 头标的生成。但它不保证 Expires 头标将被生成。如果不满足准则，就不发送头标。

浏览器提醒您：
使用本软件
请尊重相关知识产权

5.16.2 ExpiresByType

语　法 1：ExpiresByType MIME-type M<seconds> | A<seconds>

语　法 2：ExpiresByType MIME-type "<base time> [plus] <num>
<years|months|weeks|days|hours|minutes|seconds>"

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

此命令指定用于指定的 MIME 类型文档的 Expires HTTP 头标的值。过期时间以秒为单位。可以定义在两天内的任意时间。如果选择使用 M<seconds>格式来指定过期时间，那么该文件的最后修改时间被用作时间基数。换句话说，M3600 表示想要文件在它被最后修改后的 1 小时后过期（无效）。而另一方面，如果使用了 A<seconds>格式，那么客户端的访问时间就被用作时间基数。下面是一些例子。

下面的语句使所有纯文本文件在客户端缓存一小时后过期：

```
ExpiresByType text/plain A3600
```

下面的语句使所有的 GIF 文件从最后修改时间算起一个星期后过期：

```
ExpiresByType image/gif M604800
```

如果需要将第二种语法用于指定过期时间，则需要使用下面的选项来确定合适的<base time>（时间基数）值：

- Access——客户访问该文件的时间。
- Now——当前时间。它与访问时间是相同的。
- Modification——文件被最后改动的时间。

例如：

```
ExpiresByType text/html "access plus 7 days"
ExpireBsyType image/gif "modification plus 3 hours 10 minutes"
```

5.16.3 ExpiresDefault

语　法 1：ExpiresDefault M<seconds> | A<seconds>

语　法 2：ExpiresDefault "<base time> [plus] <num>
<years|months|weeks|days|hours|minutes|seconds>"

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：Indexes

此命令为指定的使用范围内的所有文档设置缺省的过期时间。例如，如果在虚拟主机使用范围内指定此命令，它将只适用于可以通过虚拟主机访问的文档。与此类似，可以在针对目录的使用范围内指定此命令，这将使该目录中的所有文档以指定的时间间隔过期（失效）。有关语法的细节内容，请参见 ExpiresByType。这里有一些例子：

使用本复制品请尊重相关知识产权!

♦ExpiresDefault　M3600

♦ExpiresDefault　"access　plus　2 day"

第一个例子将过期时间设置为在文档的最后修改时间后的 1 小时。第二个例子将过期时间设置为客户访问后的 2 天。

5.17　mod_headers

此模块在缺省情况下不被编译。它能够操作 HTTP 应答头标，还提供了单独的名为 Header 的命令。

5.17.1　Header

语　法 1：Header　<action>　<header>　<value>

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：FileInfo

此命令能够控制 HTTP 应答头标。允许的操作有：

♦Set——设置头标。如果存在相同名称的旧的头标，则用新值代替它的旧值。

♦Add——添加头标。当一个或多个具有相同名称的头标存在时，这可以产生多个具有相同名称的头标。

♦Append——将值添加到已存在的头标值中。

♦Unset——删除头标。

例如：

```
Header　add　Author　"Mohammed J. Kabir"
```

此行语句添加具有“Mohammed J. Kabir”值的 Author 头标。而下行则删除此头标：

```
Header　unset　Author
```

5.18　mod_imap

在缺省情况下，此模块在 Apache 中被编译。它提供图形映射支持，而此功能以前由称为 imagemap 的 CGI 程序提供。你可以用 AddHandler 命令来为任何文件扩展名指定 imap-file 处理程序（建在此模块中）。例如：

```
AddHandler　imap-file　map
```

这使得 Apache 将所有带有 .map 扩展名的文件作为图像映射文件。Apache 使用此模块来处理这一文件。注意，旧的格式为：

```
AddType  application/x-httpd-imap  map
```

此模块仍然支持旧的格式；然而，建议不要使用它。在图像映射文件中的行可以具有以下格式中的任一种：

```
directive  value  [x,y...]
directive  value  "Menu  text"  [x,y...]
directive  value  x,y... "Menu text"
```

超星阅览器提醒您：
作品本复制品
请尊重相关知识产权！

允许使用的命令有：

♦base——在映射文件中使用的相对 URL 被认为是相对于此命令的值。注意，当在映射文件中出现此命令时，则此命令覆盖由 Imapbase 命令设置的值。它的缺省值为 http://server_name。base_uri 与 base 同义。

♦default——当坐标不适合于启动任何 poly、circle 或 rect 命令，而且又没有给出 point 命令时指定所采取的操作。此命令的缺省值为 nocontent，它告诉客户保持显示同一页。

♦poly——使用最少三个点、最多一百个点定义一个多边形。如果用户提供的坐标落入多边形范围，则启动此命令。

♦circle——使用中心坐标和在圆上的一点定义一个圆。如果用户提供的坐标落入圆的范围，则启动此命令。

♦rect——使用两个对角的坐标定义一个矩形。如果用户提供的坐标落入矩形的范围，则启动此命令。

♦point——定义单个点的坐标。当没有其他的命令被满足时，最接近用户提供的坐标的 point 命令被使用。

命令值（value）是绝对或相对的 URL，或者是在后面列表中的特殊值之一。坐标（x，y）用空白字符隔开。如果生成图像映射菜单，双引号内的正文用作链接文本。任何以 # 字符开头的行都认为是注释，并被忽略。

特殊的图像映射文件命令值有：

♦URL——一个绝对的或相对的 URL。相对的 URL 相对于 base 来进行解析。

♦Map——与图像映射文件本身的 URL 相同。除非 ImapMenu 被设置为 none，否则将创建一个菜单。

♦Menu——与 Map 相同。

♦Referer——与引用文档的 URL 相同。如果 Referer:头标不存在，则它缺省值为 http://servername/。

♦Nocontent——发送状态码 204，它告诉客户端保持显示同一个页面。此值对 base 无效。

♦Error——发送状态码 500，通知客户端出现服务器错误。

以 x，y 格式写出坐标，其中每个坐标用空白字符分开。当生成菜单时，引号内的正文将用作链接正文。在没有这样的正文时，在下面的图像映射文件例子中 URL 将是所显示的链接：

```
# Comments goes here
# Version 1.0.0
```

```
base  http://www.yoursite.com/some/dir
rect  thisfile.html  "Customer info"  0,0 100,200
circle http://download.yoursite.com/index.html 295,0 100,22
```

如果此图像映射文件名为 imagemap.map，则可以从另一个 HTML 文件中用如下的方式来引用它，例如：

```
<A HREF="/path/to/imagemap.map"> <IMG  ISMAP
SRC="/path/to/imagemap.gif"></A>
```

5.18.1 ImapMenu

超星阅览器提醒您：
使用本复制品
请尊重知识产权！

语　　法：ImapMenu {none, formatted, semi-formatted, unformatted}

使用范围：服务器配置，虚拟主机，目录，.htaccess

覆　　盖：Indexes

在遇到请求图像映射文件而没有提供任何有效坐标的情况时，此命令确定应采取的操作。它允许以下的操作：

♦none——不生成菜单，执行缺省的操作。

♦Formatted——生成最简单的菜单。注释被忽略。打印第一层头标，随后是 hrule，最后是链接——每一个都在分开的行上。

♦Semiformatted——在半格式化的菜单中打印注释，而空行被转换成 HTML break。不打印头标或 hrule。

♦Unformatted——在无格式的菜单中打印注释，而忽略空行。

5.18.2 ImapDefault

语　　法：ImapDefault {error, nocontent, map, referer, URL}

使用范围：服务器配置，虚拟主机，目录，.htaccess

覆　　盖：Indexes

此命令定义用于图像映射的缺省操作。通过在图像映射文件中使用 default 命令，可以覆盖此缺省设置。

5.18.3 ImapBase

语　　法：ImapBase {map, referer, URL}

使用范围：服务器配置，虚拟主机，目录，.htaccess

覆　　盖：Indexes

此命令设置在图像映射文件中的缺省 base 值。通过在图像映射文件中使用 base 命令，可以覆盖此 base 值的设置。如果此命令不存在，则缺省的 base 值为 http://servername/。

5.19 mod_include

请参见第 7 章。

5.20 mod_info

请参见第 11 章。

5.21 mod_log_agent

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

请参见第 11 章。

5.22 mod_log_config

请参见第 11 章。

5.23 mod_log_referer

请参见第 11 章。

5.24 mod_mime

在缺省情况下，此模块被编译。它用来向客户端提供有关文档的元信息。它还能够定义用于文档的处理程序，以确定 Apache 如何处理此文档。

5.24.1 AddEncoding

语　　法：`AddEncoding MIME-encoding file-extension file-extension...`

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（`.htaccess`）

覆　　盖：`FileInfo`

此命令将一个或多个文件扩展名映射到一个 MIME-encoding 方案。例如：

```
AddEncoding  x-gzip  gz
AddEncoding  x-tar  tar
```

这将导致名为 backup.gz 的文件被映射为 x-gzip 编码文件，而名为 tarball.tar 的文件被映射为 x-tar 编码文件。

5.24.2 AddHandler

语　　法：AddHandler handler-name file-extension file-extension...
使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）
此命令被用来为一个或多个文件扩展名定义一个处理程序。例如：

```
AddHandler server-parsed .shtml
```

这指定所有的.shtml 文件将由名为 server-parsed 的处理程序处理，该程序可以在 mod_include（请参见第 7 章）中找到。

5.24.3 AddLanguage

语　　法：AddLanguage MIMI-language file-extension file-extension...
使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）
覆　　盖：FileInfo
此命令将文件扩展名列表映射到 MIME-language。例如：

```
AddLanguage en .en .english
```

此命令将所有带.en 和.english 后缀的文件映射为英语文件。这在内容协商中很有用。其中服务器可以根据客户喜欢使用的语言返回文档。例如：

```
AddLanguage en .en
AddLanguage fr .fr
```

如果客户喜欢英语文档，而且 document.fr.html 和 document.en.html 都可以得到，则服务器应该返回 document.en.html。

5.24.4 AddType

语　　法：AddType MIME-type file-extension file-extension...
使用范围：服务器配置，虚拟主机，目录， 针对目录的配置文件（.htaccess）
覆　　盖：FileInfo
此命令将文件扩展名列表映射到一个 MIME-type。例如：

```
AddType text/html htm html. HTM HTML
```

5.24.5 ForceType

语　　法：ForceType MIMI-type
使用范围：目录，针对目录的配置文件（.htaccess）
此命令用来对目录内所有文件强制实施一定的 MIMI 类型。目录可以由<Directory>或<Location>容器来指定。例如：

```
<Directory /www/nitec/public/htdocs/files/with/no/extensions>
ForceType text/html
</Directory>
```

这将对在指定目录内的所有文件强制实施 text/html MIME 类型，不管这些文件的原

来的扩展名是什么。

5.24.6 SetHandler

超星阅览器提醒您：
使用本复制品
需尊重相关知识产权！

语　　法：SetHandler handler-name

使用范围：目录，针对目录的配置文件（.htaccess）

此命令用来为目录或 URL 位置定义处理程序。例如：

```
<Location /ssi>
SetHandler server-parsed
</Location>
```

这一设置强制将所有在/ssi 位置中的文件作为 SSI 看待，它们由 server-parsed 处理程序来处理。

5.24.7 TypesConfig

语　　法：TypesConfig filename

缺 省 值：TypesConfig conf/mime.types

使用范围：服务器配置

此命令指定缺省的 MIME 配置文件。对于大多数 Apache 安装来说，缺省值应该很合适。如果想加入自己的 MIME 类型，则请使用 AddType 命令，而不是修改此文件。

Tip

如果需要对处理 MIME 类型增加额外的支持，需要查看 mod_mime_magic 模块的说明。对于大多数 Apache 来说安装它并不是必要的，因此，在本书中没有讲解这方面的内容。

5.25 mod_negotiation

在缺省情况下此模块被编译。它提供了对内容协商的支持。在典型的内容协商方案中，客户端提供有关它能够处理什么类型内容的信息，而服务器则尝试提供最合适的内容。服务器在类型映射和 MultiViews 搜索的帮助下完成此项功能。

类型映射提供了文档的描述信息。每个文档描述中包含一个或多个头标。它还可以包含以 # 字符开头的注释行。文档描述用空白行分开。这些文档描述头标是：

♦Content-Encoding——它指定了文件的编码类型。当前只允许使用 x-compress 和 x-gzip 编码。

♦Content-Language——文档使用的语言。

♦Content-Length——以字节计算的文件长度。

♦Content-Type——文档的 MIME 类型。允许使用可选的 key=value 参数。所允许的参数有 level 和 qs。第一个参数提供了 MIME 类型的版本号（是一个整数），而第二个参数被用来指示文档的质量（是一个浮点数）。

♦URI——到文档的相对于映射文件的路径。

MultiViews 搜索试图使用从客户端得到的信息来确定与缺少的文档最接近的匹配，

并在可能的情况下返回匹配的文档。当在 Options 命令中使能 MultiViews 选项时，服务器在未找到被请求的文档时执行 MultiViews 搜索。

Mod_negotiation 提供了下面两个命令。

5.25.1　CacheNegotiatedDocs

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

语　　法：CacheNegotiatedDocs

使用范围：服务器配置

此命令允许代理服务器缓存内容协商的文档。注意，新的 HTTP/1.1 规范对于缓存协商文档提供更好的支持，而 CacheNegotiatedDocs 对于响应 HTTP/1.1 请求没有影响。在 HTTP/1.1 被广泛使用后，此命令很可能会消失。不建议使用 CacheNegotiatedDocs。

5.25.2　LanguagePriority

语　　法：LanguagePriority　MIME-language　MIME-language...

使用范围：服务器配置，虚拟主机，目录，针对目录的配置文件（.htaccess）

覆　　盖：FileInfo

当客户端没有提供语言偏好信息时，此命令指定服务器在 MultiViews 搜索方案中应该使用什么样的语言顺序。例如：

```
LanguagePriority en fr de
```

如果打开了 MultiViews 选项，并且客户端没有对缺少的文件提供语言偏好信息，则服务器首先尝试提供最接近匹配的英语版本，随后是法语版本，依次类推。与 CacheNegotiatedDocs 类似，此命令在 HTTP/1.1 环境中不起作用。

5.26　mod_rewrite

请参见第 18 章。

5.27　mod_setenvif

在缺省情况下，此模块在 Apache 中被编译。它使你能够创建定制环境变量，而这些变量可以在以后被用来做决定。

5.27.1　BrowserMatch

语　　法：BrowserMatch　regex　variable[=value]　[...]

缺 省 值：none

使用范围：服务器配置

覆　　盖：none

此命令用来在规则表达式与模式匹配时设置和删除定制环境变量。例如：

```
BrowserMatch ^Mozilla vbscript=no javascript
```

如果 User-Agent HTTP 请求头标字段包含关键字 Mozilla，因为没有为名为 javascript 的环境变量指定值，则将该变量设置为 1，并将名为 vbscript 的变量设置为值 no。让我们看另外一个例子：

```
BrowserMatch IE vbscript !javascript
```

在这里，如果在 User-Agent HTTP 请求头标中发现了关键字 IE，则删除变量 javascript，并设置 vbscript 为 1。

注意，规则表达式的匹配与大小写有关。

5.27.2 BrowserMatchNoCase

语　　法：`BrowserMatchNoCase regex variable[=value] [...]`

缺 省 值：none

使用范围：服务器配置

覆　　盖：none

除了为规则表达式提供与大小写无关的匹配外，此命令和 BrowserMatch 命令相同。例如：

```
BrowserMatchNoCase ^MSIE vbscript=yes
```

这将匹配 MSIE、msie、Msie 等等。

5.27.3 SetEnvIf

语　　法：`SetEnvIf attribute regex envar[=value] [...]`

缺 省 值：none

使用范围：服务器配置

覆　　盖：none

和 BrowserMatch 以及 BrowserMatchNoCase 命令类似，此命令能够设置和删除定制的环境变量。实际上，BrowserMatch 和 BrowserMatchNoCase 是 SetEnvIf 的两种特殊形式。这两个命令只能在 User-Agent HTTP 的请求头标字段上执行规则表达式，而 SetEnvIf 可以被用于所有的请求头标字段。例如：

```
SetEnvIf Remove_Host "yourdomain\.com" local_user=true
```

如果能够得到的话，可以使用的一些其他变量有 Remote_Addr、Remote_User、Request_Method、Request_URI 以及 Referer。

5.27.4 SetEnvIfNoCase

语　　法：`SetEnvIfNoCase attribute regex variable[=value] [...]`

缺 省 值：none

使用范围：服务器配置

覆　　盖：none

它与 SetEnvIf 相同，但是它提供了与大小写无关的规则表达式匹配。

5.28　mod_speling

在缺省情况下此模块在 Apache 中不被编译。它使你能够处理含有错误拼写的或错误大小写的 URL 请求。它将被请求文档（含有错误拼写或错误大小写）与被请求目录中的所有文档名称比较，以寻找一个或多个最接近的匹配。

在出现含有错误拼写的文档请求的情况下，该模块允许出现一个单独的拼写错误，例如插入一个多余的字符、遗漏一个字符或一次调换位置。在出现含有错误大小写的文档请求的情况下，它执行与大小写无关的文件名比较。在上面的两种方式中，如果模块找到一个与请求最相似的文档，它都发送一个重定向请求给客户端。如果有多于一个的匹配，它会发送列表给客户端以供选择。

此模块所提供的唯一命令为 CheckSpelling。

5.28.1　CheckSpelling

语　　法：`CheckSpelling　on | off`

缺 省 值：`CheckSpelling　off`

使用范围：服务器配置，虚拟主机

此命令允许或禁止使用该模块。注意，当拼写纠正被打开时，由于服务拼错文档请求需要进行额外的搜索，服务器可能经受性能损失。还应该注意，此模块只能和文件及目录名称一起使用。

5.29　mod_status

请参见第 11 章。

5.30　mod_unique_id

此模块为每个请求提供在非常特殊的条件下保证是唯一的有魅力的标识符。此唯一的标识符甚至在正确配置的机器群的多个机器中也是唯一的。环境变量 UNIQUE_ID 被设置为每个请求的标识符。

5.31　mod_usertrack

请参见第 11 章。

第二部分　管理 Web 站点

第 6 章　建立虚拟站点
第 7 章　Apache 的服务器端包含命令
第 8 章　CGI 配置
第 9 章　FastCGI

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

第 6 章　建立虚拟站点

本章内容：

- ♦你需要虚拟 Web 站点吗
- ♦对虚拟 Web 站点的 DNS 配置
- ♦怎样建立虚拟 Web 站点
- ♦虚拟主机的例子
- ♦对虚拟主机的 SMTP 邮件服务器配置

在第 3 章中，讨论了怎样配置 Apache 来建立 Web 站点。当更加熟悉 Web 后，而且个人或公司的需求改变时，可能需要更多而不只是一个 Web 站点来展示产品、服务等等。在这种情况下，需要建立运行 Apache 服务器的多个服务器机器来建立这些增加的站点。在专用的主机上运行每个 Web 站点，对新的系统管理员似乎具有吸引力，但是对那些用过一段时间和了解得更多的人们来说，他们喜欢一种称为“虚拟 Web 站点”的解决方案。本章中，你将了解到怎样用 Apache 建立虚拟 Web 站点。

6.1　你需要虚拟 Web 站点吗

在深入剖析如何在 Apache 上实现虚拟 Web 站点之前，你需要确定是否真正需要虚拟 Web 站点。这是很重要的。表 6-1 列出了虚拟 Web 方案的优缺点。

如果你认为虚拟服务器方案主要出于经济考虑，那么你与我的观点很接近。然而，从经验看还有一点必须加以考虑。许多有经验的 Web 站点管理员都知道：一般的 Web 站点既不缺乏 CPU 时间，也不缺乏资源（资源表示磁盘或内存）。因此，当他们面对一组典型的站点时，他们使用虚拟服务器代替专用服务器。然而，当预计有较高的通信流量时，选用专用服务器（每个站点可能有多个服务器）就是必然的。

表 6-1 虚拟服务器的优缺点

优　　点	缺　　点
虚拟 Web 站点要比拥有多个专用主机、在每个主机上运行单个 Web 站点的系统更易管理	每个 Web 站点专用一台主机会提高服务器的性能，因为每台主机将只对属于单个 Web 站点的调用提供服务。然而，这种性能的提高只对那些承担高通信流量的站点有好处。如果在专用 Web 服务器上运行低通信流量的站点，那么这台服务器资源的就不会得到充分利用
只有一组配置文件。只需要管理四个配置文件。在多数情况下不必改变 mime.types 文件，所以实际上只需要管理三个文件：httpd.conf、srm.conf 和 access.conf	对于每台服务器，需要管理四个配置文件，在多数情况下不必改变 mime.type 文件，但仍需保证该文件在每台服务器系统中可供使用
每当对单独的、运行一个或更多虚拟 Web 站点的服务器系统进行硬件或软件升级时，这些虚拟站点都将会从此升级中受益	当一个系统升级时，只有建立在那台服务器上的 Web 站点受益
当服务器组合（即多于一个服务器的系统）被用于建立一个或多个虚拟 Web 站点时，它们就获得了某种程度的容错能力。在一般情况下，这种类型的配置包含循环 DNS 方案及 NFS 文件系统	当建立单个 Web 站点的服务器失败，那个站点就彻底变得没用，因为其他服务器都用于不同的站点

在 Web 上的许多站点实际上是虚拟的。你能看到许多试图出售虚拟 Web 空间的广告。许多机构通过商业公司来建立他们的 Web 站点。然而，在虚拟 Web 站点成为可能之前，ISP 仍然停留在缺少专业水准的解决方案上。例如，在虚拟 Web 站点解决方案出现之前，ISP 将不得不给想拥有自己站点的人提供以下三种选择：

◆www.isp.net/customer 或 www.isp.net/~customer，它表示 ISP 的 Web 站点地址成为顾客站点地址的一部分，因而提供不很优雅的形象和感受。

◆http://<顾客的域>:<非标准端口>/ 是另一种解决办法。例如，www.gunchy.com:8080/ 可以作为 Gunchy 公司的 Web 站点。注意：在这里必须使用 80（缺省的 HTTP 端口）以外的端口地址。

◆最后一种解决方案是为顾客域使用运行 Apache 服务器的专用主机。

也许你还不明白为什么花这么多篇幅介绍这些选项，下面就是原因。

你们当中仍然有人需要实施上面列出的非虚拟的方案。在实施其中一种方案前，就应该知道所有选择。例如，假如只打算建立几个没有定制域名的用户 Web 站点，可以使用 http://your.web.site/~user 作为这些用户的主页 URL。事实上，这仍然是提供用户 Web 站点的标准途径。如果想用同一个 IP 别名作为另一个站点的别名的话，上面列出的第二项是好的选择。例如，如果需要一个开发站点，能利用它为主 Web 站点试验新技术，可以试用这种方案（第 18 章将详细讨论它）。最后一种方案对于高通信流量和重负载的站点更适合。

现在，假如你打算建立经济的、虚拟 Web 站点，本章其余部分的内容将有助于你做到这一点。

6.2 域名注册

超星阅览器提醒您：
使用本复制品
请尊重知识产权！

域名服务（DNS）是对 Internet 域名的解析。例如，当在 Web 浏览器的地址（或 URL）字段里键入 www.idgbooks.com/时，Web 浏览器就与一个 DNS 服务器（通常是你的 ISP 的 DNS 服务器）进行通信，要求它将 IP 别名 www.idgbooks.com 解析成 IP 地址 206.80.51.140。虚拟 Web 站点需要使用同样的服务。在这节中将讨论怎样做到这一点。

为了弄清楚这些相当复杂的概念，先举个例子。在这一节和下一节中，将讨论建立名为 www.milkyweb.com 的虚拟 Web 站点。实际上它被建立在称为 nitec.com 的域上。nitec.com 域已经配置了 DNS 服务器，它能解析所有在域中的主机名。

例如，如果你在浏览器上敲入 www.nitec.com/，就会看到 Nitec 的 Web 页。这表明你的浏览器能让本地 DNS 服务器解析 www.nitec.com。简单地说，nitec.com 设置得当。因此，现在就能在 Nitec 的 Apache Web 服务器上建立新站点，这台服务器在主机 www.nitec.com 上运行。

注册域名的困难：

现在要找到一个好的域名变得愈加困难了。似乎某人或某个公司注册了所有的英语单词。例子中提到的两个域 milkyweb.com 和 nitec.com 都属于不同机构的真实域。有趣的是，在编写本章时，我检查 milkyweb.com 以确保这个域并不存在；但到完成本章初稿后不久它也被注册了！我不得不申请允许在例子中使用这个名称。看起来，所能想到的任何一个域名都可能会在下决心注册它时就已被采用了。

要意识到这一点，并及时使新颖的域名注册。

注意，如果你自己（和许多 Web 管理员一样）不处理域名服务，可以跳到本章的基于 IP 的虚拟主机这一节。一般情况下，ISP 会处理所有 DNS 问题。下面的例子说明当不是这种情况时，应该做些什么。

如果 milkyweb.com 是新建的域，那么需要做的就是到位于 http://rs.internic.net/的 InterNIC Web 站点，找出相应的联机域名注册表并填好它。

然而，填表需要知道 nitec.com 的主域名和辅助域名服务器是什么，因为这些也为 milkyweb.com 提供 DNS 服务。找出域名服务器是件相当容易的事情。所需要做的就是运行称为 whois 的 TCP/IP 实用程序。在图 6-1 中，显示了 whois nitec.com 命令的输出结果。whois 实用程序是个查询程序，它将和 InterNIC 域数据库连接并查询域名。

正如所看到的，用于 nitec.com 的 DNS 服务器（ns.nitec.com、mach1.s-cc.com、news.s-cc.com）被列在输出结果中。然后，在域名注册申请中，键入作为 milkyweb.com 主 DNS 服务器的 ns.nitec.com 和作为辅助 DNS 服务器的 mach1.s-cc.com。一旦完成申请，

```
Telnet - blackhole
Connect Edit Terminal Help

   Domain Name: NITEC.COM

   Administrative Contact:
      Kabir, Mohammed  (MK607)  kabir@CSUS.EDU
      (916) 649-9626 (FAX) (916) 649-9513
   Technical Contact, Zone Contact:
      Prochaska, John  (JP1834)  john@S-CC.COM
      1+916-338-6580 (FAX) 1+916-338-6587

   Record last updated on 01-Dec-96.
   Record created on 28-Oct-95.
   Database last updated on 3-Dec-97 05:19:16 EDT.

   Domain servers in listed order:

   NS.NITEC.COM                207.212.140.213
   MACH1.S-CC.COM              207.212.154.2
   NEWS.S-CC.COM               207.212.154.4


The InterNIC Registration Services Host contains ONLY Internet Information
(Networks, ASN's, Domains, and POC's).
Please use the whois server at nic.ddn.mil for MILNET Information.
[kabir@blackhole ~]$
```

图 6-1 whois nitec.com 命令的输出结果

就需要配置 Nitec 的域名服务器来回答有关 milkyweb.com 的查询。

Note

建立新域需要向 InterNIC 付费。

6.3 对虚拟 Web 站点的 DNS 配置

由于 www.milkyweb.com 主机有个域地址 milkyweb.com（不是 nitec.com），因此必须告诉 InterNIC（域名服务器授权机构）将 milkyweb.com 指向 nitec.com。这要完成两个步骤，首先要为这个域建立相应的数据库记录，然后在基于 IP 和基于名称之间作出选择。

6.3.1 建立域名的数据库记录

建立一个虚拟主机要完成的第一步是建立域名数据库。可以建立新的域名数据库，也可以在现有的域名数据库上作修改。

6.3.1.1 建立新的域名数据库

为了使 nitec.com 的域名服务器对 milkyweb.com 提供服务，nitec.com 需要知道 milkyweb.com 域名的细节。通过使用称为 milkyweb.db（如列表 6-1 所示）和 milkyweb.rev（如列表 6-2 所示）的两个文本文件，可以建立 milkyweb.com 域名所需要的 DNS 记录。注意，这些文件用了广泛使用的 BIND 名称服务器文件格式。

列表 6-1 milkyweb.db

```
@ IN SOA milkyweb.com.    hostmaster.milkyweb.com.(
  19981110001; Serial YYYYMMDDXXX
```

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

```
  7200 ; refresh
  3600 ; (1 hour) retry
  1728000 ; (20 days) expire
  3600) ; (1 hr) minimum ttl

; Name Servers
IN NS      ns.nitec.com.
IN NS      mach1.s-cc.com.

; A Records
www IN A   206.171.50.50
```

尽管对所有 DNS 记录进行完整解释超出了本书范围，但讲述了应该了解的最少内容。第一个记录是 SOA 记录，它指定什么时候远程 DNS 服务器应该刷新、重试和使提供的信息过期。这使 DNS 记录提供者修改该消息并让它以某种方式在 Internet 上传播。这个记录还包含没有@符号的电子邮件地址（如 hostmaster.milkyweb.com）和一个系列号。电子邮件地址用于管理的目的，而系列号只是为记录保存用的。我喜欢用特殊的 YYYYMMDDXXX 格式，在这里我用月、日、年和一个单独的 3 位数来建立一个系列号。DNS 服务器查寻这个系列号，检查服务器保存的有关域名的信息是否需要进行修改。因此，每次修改 DNS 记录时，都应该增加这个数。

下一组记录是 DNS 记录，它指定由哪些域名服务器给这一域名提供 DNS 服务。在这种情况下，milkyweb.com 的 DNS 服务器实际上和用于 nitec.com 的 DNS 服务器一样。

Tip　DNS 管理新手常犯的错误是忘记用句号结束主机别名（如 ns.nitec.com）。

最后一个记录是 A 记录或地址记录。这个记录指定主机和地址之间的对应关系。可以用缩写方式，其中 www 就是 www.milkyweb.com 的缩写。这正是你所想得到的。若想建立称为 www.milkyweb.com 的主机，就需要用前面的所有设置来告诉世界各地的人们 www.milkyweb.com 是在 milkyweb.com 域中的主机，它被建立在 nitec.com 域中，并且它的主域名服务是由 ns.nitec.com 提供的。

Note　我这里用的 IP 地址并非胡诌的，它是 nitec.com 的有效的 IP 地址，并且和这个 IP 地址相连接的机器会服务于 www.milkyweb.com。对 DNS 的设置过程还没有结束。你仍需要给其他计算机提供一种方法来做逆向名称查找。因此，需要建立如列表 6-2 所示的 milkyweb.rev 数据库文件。

列表 6-2 milkyweb.rev

盗版警告 使用本复制品 请尊重相关知识产权!

```
@ IN SOA milkyweb.com.    hostmaster.milkyweb.com.(
   19981110001 ; Serial YYYYMMDDXXX
   7200 ; refresh
   3600 ; (1 hour) retry
   1728000 ; (20 days) expire
   3600) ; (1 hr) minimum TTL

; Name Servers
IN NS      ns.nitec.com.
;  PTR Records
50 IN PTR www.milkyweb.com.
```

这个文件和列表 6-1 中的文件有一样的 SOA 和 DNS 记录。在这个文件中只有一个新记录，即 PTR 记录，它将 IP 地址指向 IP 别名。在这种情况下，我只用这个 PTR 记录中的 IP 地址（206.171.50.50）的主机部分（C 类网络的最后一个 8 位字节）将它指向 www.milkyweb.com。

现在，如果 Internet 上的服务器有兴趣了解 206.171.50.50 是否真正是 www.milkyweb.com，那么它可以做逆向名称查找来确认这点。

现在需要在 nitec.com 域的名称服务器配置文件里使用这两个文件，它存储在 /etc/named.boot 里。在这个配置文件中，需要添加的几行如下：

```
primary     milkyweb.com              milkyweb.db
primary     50.171.206.in-addr.arpa   milkyweb.rev
```

第一行表明这台 DNS 服务器（ns.nitec.com）也是 milkyweb.com 的主 DNS 服务器，并且这些记录存放在 milkyweb.db 文件中。第二行说明主逆向查找记录存储在 milkyweb.rev 文件里。注意：逆向 DNS 通常是由 ISP 来处理的。

一旦这几行被添加到/etc/named.boot 文件中，需要重启动名称服务器进程，使它有机会读这些文件。这是通过下面的命令来完成的：

```
/usr/sbin/named.restart
```

一般情况下，你必须作为 root 用户来做前面的工作。这个命令实际上是对一个 shell 脚本的调用，这个脚本执行带有 named restart 参数的/usr/sbin/ndc 命令。ndc 程序是对名称服务器的接口程序。然而名称服务器软件可以用不同的程序或选择来重启动它。参阅名称服务器手册以取得帮助。

一旦重启动名称服务器后，需要对它进行测试。在图 6-2 中，当在 shell 提示符后敲入以下命令时：

```
ping  www.milkyweb.com
```

从 www.milkyweb.com 主机得到的响应。

该测试应足以令你信服 DNS 配置正在起作用。然而，我们将讨论更可靠的检测 DNS 配置的方法。因为万一配置有误，ping 只是不起作用而已，并不提供任何线索。因此，

```
Telnet - blackhole
Connect Edit Terminal Help
[kabir@blackhole ~]$ ping www.milkyweb.com
PING blackhole.nitec.com (206.171.50.50): 56 data bytes
64 bytes from 206.171.50.50: icmp_seq=0 ttl=64 time=0.3 ms
64 bytes from 206.171.50.50: icmp_seq=1 ttl=64 time=0.2 ms
64 bytes from 206.171.50.50: icmp_seq=2 ttl=64 time=0.2 ms
64 bytes from 206.171.50.50: icmp_seq=3 ttl=64 time=0.2 ms
64 bytes from 206.171.50.50: icmp_seq=4 ttl=64 time=0.2 ms

--- blackhole.nitec.com ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.3 ms
[kabir@blackhole ~]$
```

图 6-2　ping www.milkyweb.com

让我们看一看被称作 nslookup 的名称服务器查找实用程序。

这个实用程序，正如它的名称所表示的，用来完成名称服务器查找。图 6-3 显示了 nslookup 对话框。在这里键入了一些 nslookup 命令以确定前面的 DNS 配置是否可行。

```
Telnet - blackhole
Connect Edit Terminal Help
[kabir@blackhole ~]$ nslookup
Default Server:  www.milkyweb.com
Address:  206.171.50.50

> set q=ns
> milkyweb.com
Server:  www.milkyweb.com
Address:  206.171.50.50

milkyweb.com    nameserver = ns.nitec.com
ns.nitec.com    internet address = 206.171.50.50
> set q=a
> www.milkyweb.com
Server:  www.milkyweb.com
Address:  206.171.50.50

Name:    blackhole.nitec.com
Address:  206.171.50.50
Aliases:  www.milkyweb.com

> exit
[kabir@blackhole ~]$
```

图 6-3　使用 nslookup 来测试 DNS 配置

在 shell 提示符后执行如下的命令：

```
nslookup
```

这将在“>”提示符后开始对话。它还显示用来完成这些查找的 DNS 服务器。如果这台 DNS 服务器不是新配置的，用如下的命令告诉 nslookup 与刚配置的服务器进行通信：

```
server <servername>
```

在这个例子中，缺省的服务器是 ns.nitec.com，它有经修改的 DNS 设置，所以不必敲入上面的命令。回到图 6-3，会看到在 nslookup 提示符下敲入的第一组命令是：

```
set q=ns
milkyweb.com
```

超星阅览器提醒您：

第一行告诉 nslookup 将查询类型设置为 NS（名称服务器）记录。而另一行只是简单地告诉它去查找 milkyweb.com 的名称服务器记录。输出结果表明，ns.nitec.com 报告说 milkyweb.com 正由 ns.nitec.com 提供域名服务。现在再看第二组 nslookup 命令：

```
set q=a
www.milkyweb.com
```

第一条命令告诉它将查询类型设置为 A（地址）记录。然后向 ns.nitec.com 询问 www.milkyweb.com 的地址。输出结果显示这个 IP 别名的地址是 206.171.50.50。

最后，通过下面这条命令退出：

```
exit
```

到此为止，对 www.milkyweb.com 的主服务器配置就完成了。然而，仍需要配置辅助 DNS 服务器。可以在每个辅助 DNS 服务器的/etc/named.boot 文件中添加下面几行以做到这点：

```
secondary   milkyweb.com                    milkyweb.db
secondary   50.171.206.in-addr.arpa         milkyweb.rev
```

注意取代下面的 A 记录：

```
www      IN      A       206.171.50.50
```

可以用下面的 CNAME 记录：

```
www      IN      CNAME   blackhole.nitec.com.
```

这有什么不同吗？嗯，不多。因为 206.171.50.50 是 blackhole.nitec.com 的真正地址。然而，由于不使用 A 记录，使 milkyweb.com 的 DNS 更加容易管理。因为每当需要改变 blackhole.nitec.com 主机的 IP 地址时，你完全消除了修改 milkyweb.com 的 DNS 记录的必要。基于 IP 的虚拟主机仍然需要 A 记录形式，并且基于名称的虚拟主机也需要 CNAME 形式。在本章后面，将讨论这两种类型的虚拟主机。

6.3.1.2 对已有的域名数据库的修改

到此为止，已讨论了为新的域名（milkyweb.com）建立虚拟主机名的有关知识。然而，如果你已有了名为 www.yourcompany.com 的 Web 站点，并且想建立名为 intranet.yourcompany.com 的虚拟站点，那么所应该做的就是在 DNS 服务器里为 www.yourcompany.com 使用 CNAME 记录建立一个别名，如下：

```
intranet     IN     CNAME     www
```

在逆向 DNS 数据库里，需要复制 www.yourcompany.com 行的条目并用 intranet 替换 www 部分。

例如，设置如下内容：

```
apache  IN  CNAME  blackhole            ;  in nitec.db  flle
50      IN  PTR    apache.nitec.com.    ;  in nitec.rev  file
```

现在，当使用 nslookup（如图 6-4 所示）检查 apache.nitec.com 的 IP 地址时，就会看到返回 blackhole.nitec.com 机器的 IP 地址，这正是所期望的。

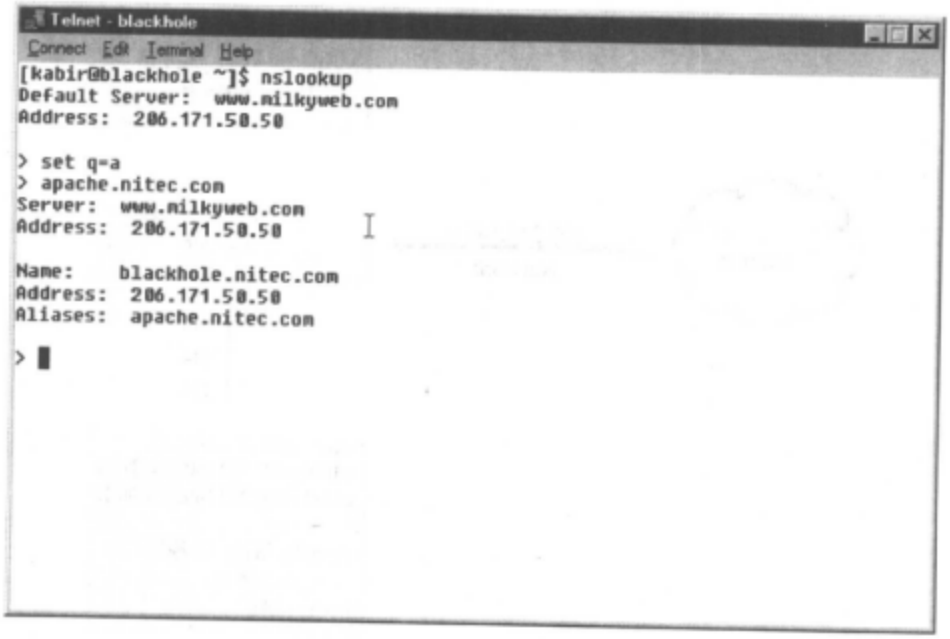

图 6-4　使用 nslookup 验证域名记录的修改

当使用基于名称的虚拟主机时，需要前面类型的配置。但如果使用基于 IP 的虚拟主机，就不得不用 A 记录代替了。例如，为完成同样的目标，可以按如下方式设置同样的主机 apache.nitec.com:

```
apache IN  A      206.171.50.50        ;  in nitec.db  flle
50     IN  PTR    apache.nitec.com.    ;  in nitec.rev  file
```

然而，这种形式可以用于基于 IP 的虚拟主机。

6.3.2　在基于 IP 和基于名称之间作出选择

在建立虚拟 Web 站点的过程中，下一步需要作出决定：要基于 IP 的虚拟主机，还是基于名称的虚拟主机。在本节中，将讨论这两种选择。

6.3.2.1　基于 IP 的虚拟主机

基于 IP 的虚拟主机表示给每个新主机一个它自己的 IP 地址。图 6-5 表示了这种配置，在这里每个虚拟主机都有它自己的 IP 地址，并且这些 IP 地址都被路由到单独的机器上。

为每个虚拟站点使用唯一的 IP 地址表示需要为每个站点建立一个 A 记录。例如，图 6-5 中所示的配置会导致如下的 DNS 配置：

```
; The following entry goes in nitec.com's DNS database
```

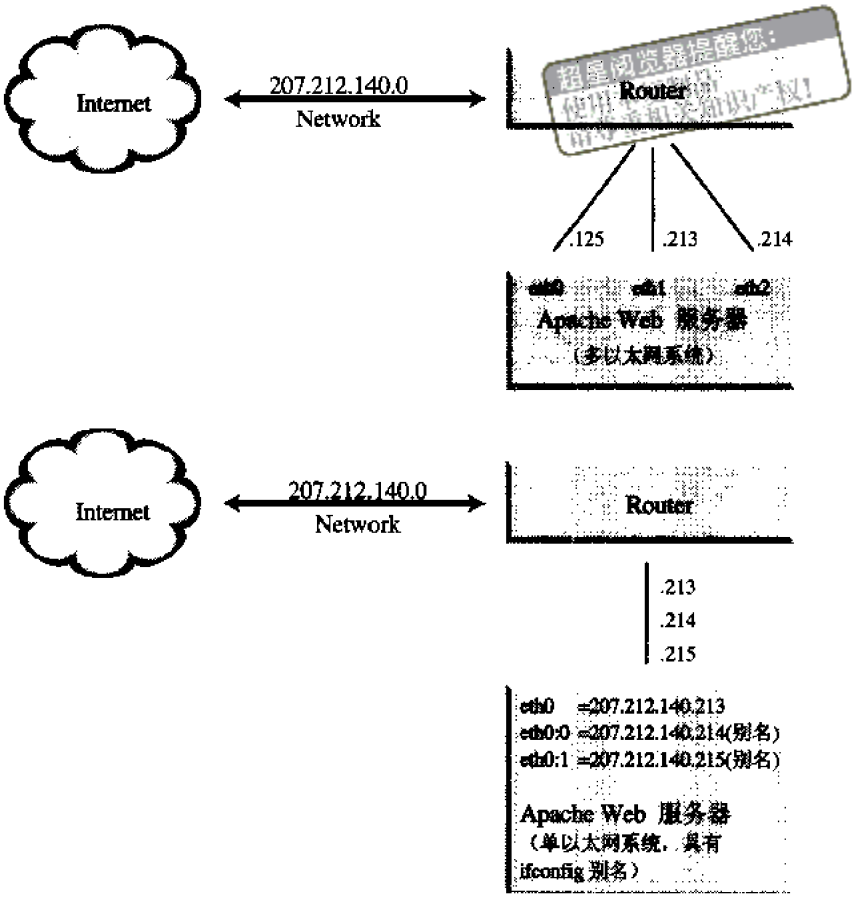

图 6-5 为每个虚拟站点使用唯一的 IP 地址

```
{ref hyperlink http:www.nitec.com}   IN   A   207.212.140.213
; The following entry goes in client1.com's DNS database
{ref hyperlink http:www.client1.com  IN   A   207.212.140.214
entry goes in client2.com's DNS database
; this
{ref hyperlink http:www.client2.com  IN   A   207.212.140.215
```

还需要设置相应的路由分配，以便所有用于 Web 服务器（虚拟的和主要的）的 IP 地址被正确地路由到目标服务器系统。取决于主 Web 服务器所用的操作系统，可以选择为单个以太网接口建立别名，或者使每个 IP 地址拥有不同的接口（一般通过给该系统添加额外的以太网卡）。

在下面例子中，将讨论怎样用别名将多个 IP 地址连接到单独的以太网设备。这个例子使用 Linux 操作系统和 ifconfig 实用程序，它能用来建立 IP 别名。如果使用不同的操作系统，请参阅相应的 ifconfig 手册，检查是否可用别名选项。

Tip

流行的操作系统，如 Linux、BSDI、Digital UNIX、FreeBSD、HPUX 10.x 及 AIX 4.1，它们在 ifconfig 实用程序中都可使用别名。然而，其中一些操作系统以及其他操作系统可能需要 ifconfig 工作所需要的内核补丁。

图 6-6 表示由 root 用户执行 ifconfig 命令后的输出结果：

图 6-6　建立别名前 ifconfig 的输出

正如从图中看到的，使用 ifconfig 已建立两个设备：本地回环设备 lo 和缺省以太网设备 eth0。现在，给 eth0 设备添加 IP 别名，需要按如下方式使用 ifconfig 命令：

```
Ifconfig device:N IP_address
```

这里，N 是 0、1、2、3 等等。例如，使用 IP 地址 206.171.50.56 为 eth0 建立别名，需要敲入如下的命令：

```
Ifconfig eth0:0 206.171.50.56
```

执行这一 ifconfig 命令会产生类似图 6-7 的输出结果。

图 6-7　建立别名后 ifconfig 的输出

有了别名的 IP 地址还需要增加如下的路由：

```
route add -host 206.171.50.56 dev eth0:0
```

在/etc/rc.d 目录下的 rc.local 文件中添加上面的命令，确保在服务器启动时这个别名被正确地设置。

因为当前的 IP 编址方案中只有有限数目的 IP 地址可供使用，所以 Internet 的 IP 地址是稀有资源。直到新的编址方案（IPNG，IP Next Generation）被广泛使用之前，获取 IP 地址仍将是愈来愈困难的事。InterNIC 通常不给个人分配 IP 地址。InterNIC 建议向 ISP 要求 IP 地址，这表示要向 ISP 交付安排到某个指定系统（如 Web 服务器）的 IP 地址的一笔费用。

与 HTTP/1.1 以前的浏览器兼容：

这个变通方案包括在第一个虚拟主机配置中使用 ServerPath 命令。配置的例子为：

```
NameVirtualHost 111.222.333.444

<VirtualHost 111.222.333.444>
#primary virtual host
DocumentRoot /www/subdomain
...
</VirtualHost>

<VirtualHost 111.222.333.444>
DocumentRoot /www/subdomain/client1
ServerName www.client1.com
ServerPath  /client1/
...
</VirtualHost>

<VirtualHost 111.222.333.444>
DocumentRoot /www/subdomain/client2
ServerName www.client2.com
ServerPath  /client2/
...
</VirtualHost>
```

在第一个虚拟主机的索引页（/www/subdomain/index.html）中，你需要用一个 URL 前缀与基于名称的虚拟主机建立连接。还需要在第一个虚拟主机的文档根目录下找到所有基于名称的虚拟主机，它们每个都在各自的子目录下。在前面例子中，第一个虚拟主机的文档根目录是/www/subdomain/。www.client1.com 和 www.client2.com，都有它们的文档根目录，分别放在/www/subdomain/client1 目录和/www/subdomain/client2 目录中。在这个虚拟 Web 站点中需要遵守的最后一条原则是所有连接都是相对连接（例如，file.html 或 ../icons/image.gif）或前面带有 /domain/ 的连接（例如，www.domain.tld/domain/misc/file.html 或/domain/misc/file.html）。

ServerPath 命令使得所有到 URL www.client1.com/client1/的请求总是由 client1 虚拟主机提供服务。如果该客户使用 HTTP/1.1 发送正确的 Host:头标，到 URL www.client1.com/的请求只由 client1 虚拟主机提供服务。如果没有发送任何 Host:头标，客户就接收来自第一个虚拟主机的信息页。

该方案看起来过于烦琐且难以管理，但坚持使用这些指导原则将在很大程度上确保你的主页适用于所有新的和旧的浏览器。

6.3.2.2　基于名称的虚拟主机

由于 HTTP/1.1 被广泛采纳，现在就能用另一种方法来建立非 IP 方案的虚拟 Web 主机。这使同一个 IP 地址可供多个主机名称使用。

在 HTTP/1.1 之前，还没有哪种方法能使 Web 服务器知道浏览器用户送入服务请求的那一台主机的名称。新的 HTTP/1.1 协议添加一项新功能：浏览器必须在“Host:头标”中告诉服务器它所使用的主机名称。当然，如果某人在使用与 HTTP/1.1 不兼容的浏览器，它就不会发送必要的Host:头标信息，服务器也就不能找到相应的站点。请参阅“与 HTTP/1.1 以前的浏览器兼容”的补充说明，了解在这种情况下怎样建立代用方案。

要支持不是基于 IP 的虚拟主机，所要做的就是在相应的 DNS 数据库中有 CNAME 记录指向带有 A 记录的已有的主机。例如，如果需要将 www.client1.com 和 www.client2.com 主机指向一台称为 webserver.nitec.com 的 Web 服务器，那就需要修改 client1.com 和 client2.com 的 DNS 数据库，分别添加如下一些条目：

```
; The following entry goes in client1.com's DNS database
{ref hyperlink http:www.client1.com} IN CNAME
webserver.nitec.com
; The following entry goes in client2.com's DNS database
{ref hyperlink http:www.client2.com} IN CNAME
webserver.nitec.com
```

本节中，讨论了为虚拟 Web 站点设置 DNS 的两种途径。在这些知识的基础上，你应该能在使用基于 IP 还是基于名称的虚拟站点这个问题上做出决定了。

现在，让我们设置 Apache，为你的虚拟主机提供 Web 服务。

6.4　用于虚拟主机的 Apache 配置

可以用两种方式配置 Apache，使它支持多个主机（主服务器和虚拟服务器）。可以运行多个 Apache 守护进程，使每个主机拥有独立的 **httpd** 守护进程；或者运行单个 Apache **httpd** 守护进程，由它支持主服务器和所有的虚拟主机。图 6-8 说明了这两种选择。

在图中可以看到，多守护进程配置有多台主服务器，而单守护进程配置只能有一台主服务器。在这两种方法中，由主服务器产生子进程为请求提供服务。例如，在多守护进程配置中，如果有对服务器 A 的请求，它将产生一个子进程，并由它为这个请求提供服务。在单个主服务器的配置中，主服务器产生子进程来为请求提供服务。其中不同之处在于：在多守护进程的配置中，用几个 Apache 守护进程来完成单守护进程方式下单个 Apache 守护进程所做的全部工作。

对于一般的需求，需要运行单独的 Apache 守护进程。然而，由于在 ServerType、User、

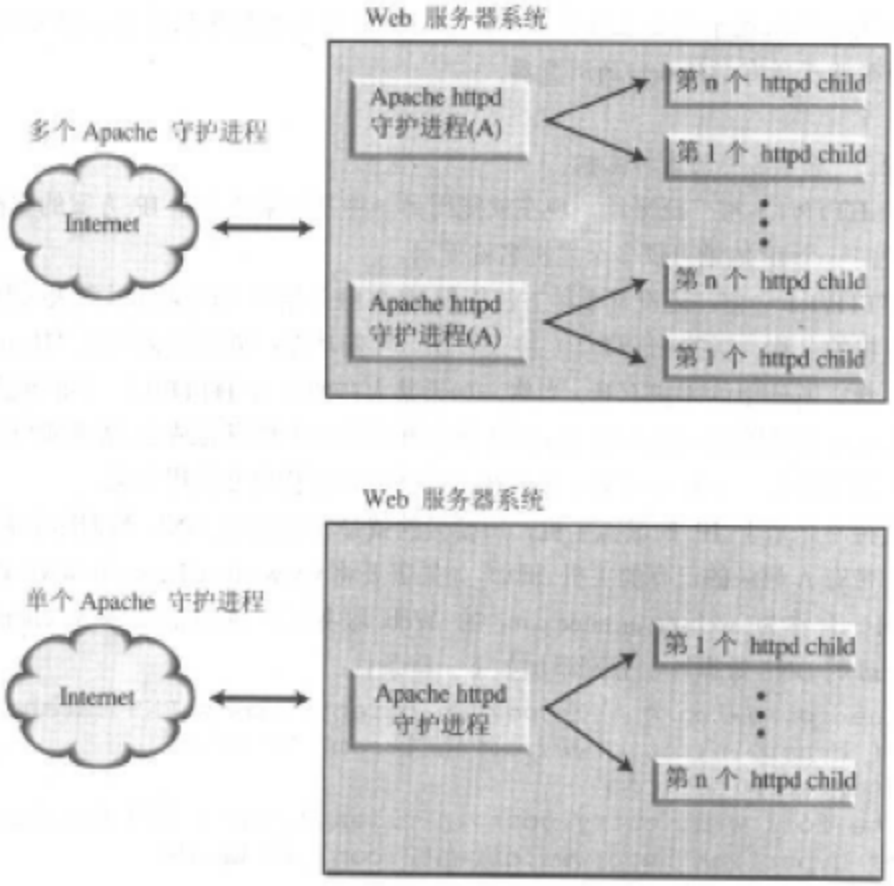

图 6-8 多个和单个 Apache 守护进程

Group、TypesConfig、ServerRoot 等等方面存在差异，在虚拟主机之间共享 httpd 配置是难以接受的。在这种情况下，必须运行多个主 Apache 进程。这些独立的主 httpd 进程将需要单独的配置文件。

让我们看看怎样配置每一种选择：设置多个守护进程和单独的守护进程。

6.4.1 设置多个 Apache 守护进程

要建立多个 Apache 守护进程，首先，必须为每个虚拟主机建立单独的 httpd 安装。基本上，需要为每个守护进程单独创建一组配置文件。对于每个安装，在配置文件中使用 Listen 命令来选择守护进程所服务的 IP 地址（或虚拟主机）。

有两种方法告诉 Apache 要侦听哪个地址和端口：或者用 BindAddress 命令指定单独的地址和端口，或者用 Listen 命令指定任意数目的特定地址或端口。例如，如果在 IP 地址 206.171.50.50 端口 80 上运行主服务器，在 IP 地址 206.171.50.54 端口 8080 上运行一台虚拟主机，那就使用：

```
Listen 206.171.50.50:80
Listen 206.171.50.54:8080
```

在 Listen 命令中用 IP 地址代替主机名较好，因为它减少了在服务器启动时将主机名解析成 IP 地址所需要的域名查寻。

超星阅览器提醒您：
使用本复制品
需尊重知识产权！

应注意的是，为虚拟 Web 站点而使用多个主服务器将加重系统的负担，所以不建议对可能承受高通信流量的站点采用这种方案。

6.4.2　设置单独的 Apache 守护进程

这是对 Apache 的缺省配置。单独的 httpd 守护进程产生子进程，它们为主 Web 站点和所有虚拟站点服务请求。在缺省的情况下，Apache 侦听本地机器上所有 IP 地址的端口 80，通常这样就足够了。如果还有更加复杂的要求，如在不同端口号上侦听或只侦听指定的 IP 地址，那就可以用 BindAddress 或 Listen 命令。

让我们看看如何将 Apache 配置成为用单独的守护进程服务所有的虚拟 Web 站点。

在 httpd.conf 文件中，Apache 用特殊容器<VirtualHost>来操纵所有特定虚拟主机的配置。下面是最小的虚拟主机配置的例子：

```
<VirtualHost 206.171.50.50>
DocumentRoot  /www/apachebible/public/htdocs
ServerName  www.apachebible.com
<VirtualHost>
```

第一行标志虚拟主机（www.idgbooks.com）配置的开始。所包括的 IP 地址需要是对 www.idgbooks.com 有效的 IP。如果需要使用非标准（即非 80）端口地址，那么可以按如下方式提供：

```
<VirtualHost IP:port>
```

例如：

```
<VirtualHost 206.171.50.50:8080>
```

你也许想知道能否用主机名代替<VirtualHost>容器内的 IP 地址。是，可以的。但在 Apache 开发人员告诉你可以这样做以前，不要这样做。如果用主机名代替 IP 地址，Apache 就必须进行一次 DNS 查找，以确定主机的 IP 地址。假如出于某种原因，DNS 查寻失败，那么这台虚拟主机可能要到重新启动 Apache 后才能使用。像许多人一样，我个人不喜欢在配置文件中保存硬编码（hard-coded）的 IP 地址，但直到 Apache 开发人员找到一种更好的处理这种情况的方法之前，仍需要使用硬编码的 IP 地址。

在<VirtualHost>容器内的任意一条命令只适用于该台虚拟主机。任何在<VirtualHost>容器外使用的命令都构成主服务器配置。除非存在冲突，每台虚拟主机继承主服务器的配置。在同一条命令同时用于主服务器配置和虚拟主机配置（就是说，用在<VirtualHost>容器内）的情况下，在虚拟主机配置中的命令就覆盖在主服务器中设置的命令。这种覆盖只对使用相同命令的特定的虚拟主机有效。例如，如果在主服务器配置中将 ServerName 设

置为 www.yourcompany.com（即是说，在<VirtualHost>容器外），同时在<VirtualHost>容器内将 ServerName 命令设置为 www.yourclient.com。你显然想使这台虚拟主机对 www.yourclient.com 的请求作出响应，对吗？那确实是所要发生的事情。在这台虚拟主机段中的这条命令覆盖了主服务器配置中的相应的命令。可以更方便地认为主服务器配置是对所有使用<VirtualHost>建立的虚拟主机的总的缺省配置。所以，当配置虚拟主机时，需要决定在每台虚拟主机配置中需要做些什么改动。

在虚拟主机配置中使用的命令或者覆盖主服务器中给出的配置，或者对它进行补充，将根据具体的命令而定。例如，在<VirtualHost>段内的 DocumentRoot 命令覆盖主服务器的 DocumentRoot，而 AddType 命令则补充主服务器的 MIME 类型。

现在，当一条请求到来时，Apache 使用请求中包含的 IP 地址和端口来查找相匹配的虚拟主机配置。如果没有虚拟主机和这个地址和端口匹配，那就用主服务器配置来处理这一请求。如果它和一台虚拟主机的地址匹配，Apache 就使用那台虚拟服务器的配置来处理这一请求。

在前面的例子中，除了 DocumentRoot 被设置为/www/apachebible/public/htdocs、ServerName 被设置为 www.apachebible.com 以外，虚拟主机所用的配置和主服务器的一样。通常设置在<VirtualHost>容器内的命令有 DocumentRoot、ServerName、ErrorLog 和 TransferLog 等等。

除了 ServerType、StartServers、MaxSpareServers、MinSpareServers、MaxRequestsPerChild、BindAddress、Listen、PidFile、TypesConfig、ServerRoot 和 NameVirtualHost 命令外，几乎可以将任何配置命令放在 VirtualHost 容器内。如果使用了 suEXEC wrapper（查阅第 12 章），User 和 Group 命令可以用于 VirtualHost 容器内。

可以使用尽可能多的<VirtualHost>命令。可以选择拥有一个或更多的由主服务器处理的虚拟主机，或拥有对每个可用地址和端口进行处理的<VirtualHost>，并使主服务器不处理任何请求。

现在，你可能想知道如果有多个非基于 IP 的虚拟主机正在使用同一个 IP 地址，Apache 如何知道应该使用哪个主机配置。Apache 足够机灵，它查找 ServerName 命令，同时在与 HTTP/1.1 相兼容的浏览器发送的 Host:头标中查找所请求的主机，并相匹配。通过将 ServerName 与请求的主机匹配，它就能决定请求哪个非 IP 主机，然后由它来服务请求。

注意 Apache 1.3 或以上的版本需要使用额外的 NameVirtualHost 命令，它应该放在实际指定基于名称的虚拟主机配置的<VirtualHost>命令之前。例如：

```
NameVirtualHost 206.171.50.50
<VirtualHost 206.171.50.50>
ServerName www.client1.com
DocumentRoot /www/client1
</VirtualHost>

<VirtualHost 206.171.50.50>
ServerName www.client2.com
```

```
DocumentRoot /www/client2
</VirtualHost>

NameVirtualHost 206.171.50.56

<VirtualHost 206.171.50.56>
ServerName www.client3.com
DocumentRoot /www/client3
</VirtualHost>
```

超星阅览器提醒您：
使用本复制品
请尊重相关知识产权！

NameVirtualHost 命令指定 IP 地址，它应该用于作为对象的基于名称的虚拟主机。在前面的例子中有两个虚拟主机，www.client1.com 和 www.client2.com，这两个对象都使用同一个 IP 地址（206.171.50.50）。第三个虚拟主机，www.client3.com 则使用不同的 IP 地址（206.171.50.56），所以第二个 NameVirtualHost 命令被正好放在这个虚拟主机配置的前面。如果按实际使用的 IP 地址将它们分组，正如在这个例子中出现的一样，对所有非 IP 虚拟主机配置只需要一个单独的 NameVirtualHost 命令。

一旦为 Apache 建立了虚拟主机配置，就能启动（或重启动）Apache 服务器以便这个配置能起作用。别忘了在进行最后一步之前建立相应的目录结构。

正如前面提到的，某些客户仍然不能发送使基于名称的虚拟主机正常工作所需的数据。这些客户常常从指定 IP 地址的配置文件中的第一个虚拟主机得到返回的页面。在下面的这节中，我将讨论通常用来解决此问题的代替方案。

6.5　一般虚拟主机的例子

现在你知道了怎样设置 DNS 和为虚拟站点配置 Apache，让我们再看一些一般的例子，这有助于理解建立虚拟 Web 站点的不同方面。

Note　在下面例子中的 IP 地址（111.222.333.x）是有意使用的无效地址。

6.5.1　在有多个 IP 地址的服务器上的基于 IP 的虚拟主机

在这个例子中，你需要配置基于 IP 的虚拟主机 www.myclient.com。它在一台名为 webserver.mycompany.com 的主机上。这台主机同时需要建立名为 www.mycompany.com 的主服务器。图 6-9 显示了这个例子所讨论的逻辑图。正如你所看到的，这个 Web 服务器有两个 IP 地址，它们有如下记录：

```
www.mywebserver.com    IN    A       111.222.333.1
www.myclient.com       IN    A       111.222.333.2
www.mycompany.com      IN    CNAME   webserver.mycompany.com.
```

简化的 httpd.conf 配置文件如下：

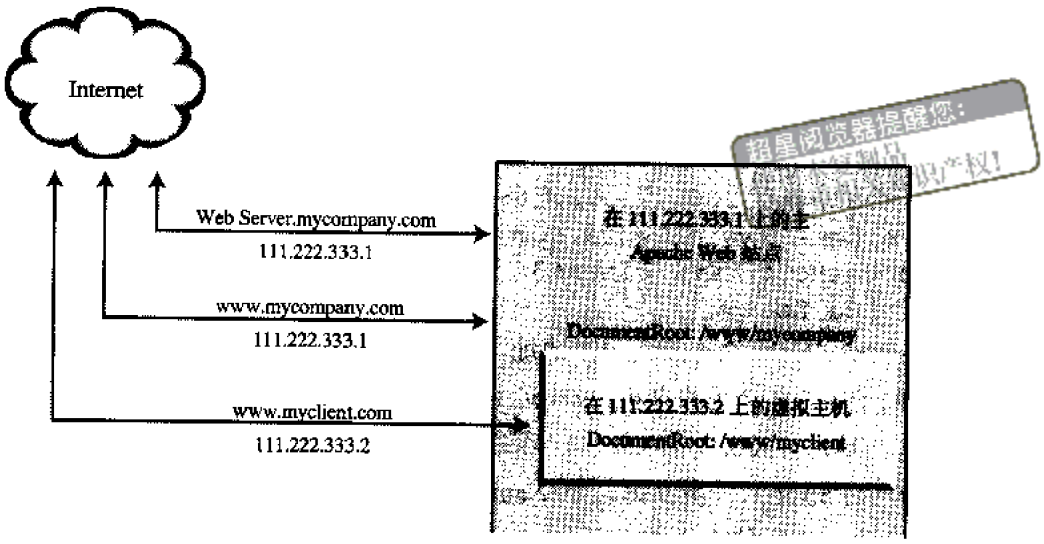

图 6-9 在有多个 IP 地址的服务器上的基于 IP 虚拟主机的例子

简化的 httpd.conf 配置文件如下：

```
...
Port 80
DocumentRoot  /www/mycompany
ServerName  www.mycompany.com

<VirtualHost 111.222.333.2>
DocumentRoot  /www/myclient
ServerName  www.myclient.com
...
</VirtualHost>
```

主机 www.myclient.com 只能通过 111.222.333.2 地址进行访问，而服务器 www.mycompany.com 只能通过 111.222.333.1 地址进行访问。

6.5.2 无主服务器的基于 IP 的虚拟主机

这个例子和前面的例子相似，所不同的在于只有本地用户（即在 webserver.mycompany.com 机器上运行他们自己的 Web 浏览器的用户）才能看到主服务器的页面。换句话说，不存在专门的主服务器。图 6-10 表示了这个例子的逻辑图。

简化的 httpd.conf 配置文件如下：

```
...
Port 80
ServerName  webserver.mycompany.com
DocumentRoot  /www/localinfo

<VirtualHost 111.222.333.1>
DocumentRoot  /www/mycompany
ServerName  www.mycompany.com
```

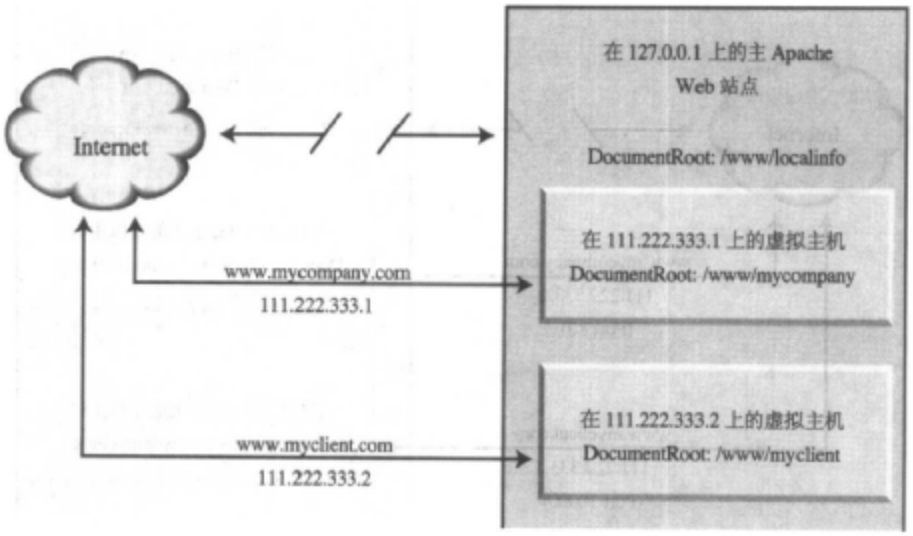

图 6-10　无主服务器的基于 IP 虚拟主机的例子

```
    ...
</VirtualHost>

<VirtualHost 111.222.333.2>
DocumentRoot  /www/myclient
ServerName  www.myclient.com
...
</VirtualHost>
```

在这种设置中，主服务器只能服务于那些正在这台机器上运行自己的 Web 浏览器的本地用户的请求。所以，当一个本地用户键入 http://localhost/时，他或她将能访问存储在 /www/localinfo 中的主服务器的页面。然而，诸如访问 www.mycompany.com/和 www.myclient.com/的请求，由它们各自的虚拟服务器提供服务。

6.5.3　不同端口上基于 IP 的虚拟主机

这个例子使用和前面两个例子一样的主机，但这次将说明怎样才能将 Apache 配置成一台侦听一个端口的虚拟主机，而另外一台侦听另外一个端口。图 6-11 显示了这个例子的逻辑图。

简化的 httpd.conf 配置文件如下：

```
...
Listen 111.222.333.1:80
Listen 111.222.333.2:10000
ServerName webserver.mycompany.com
DocumentRoot /www/localinfo

<VirtualHost 111.222.333.1>
DocumentRoot /www/mycompany
ServerName www.mycompany.com
```

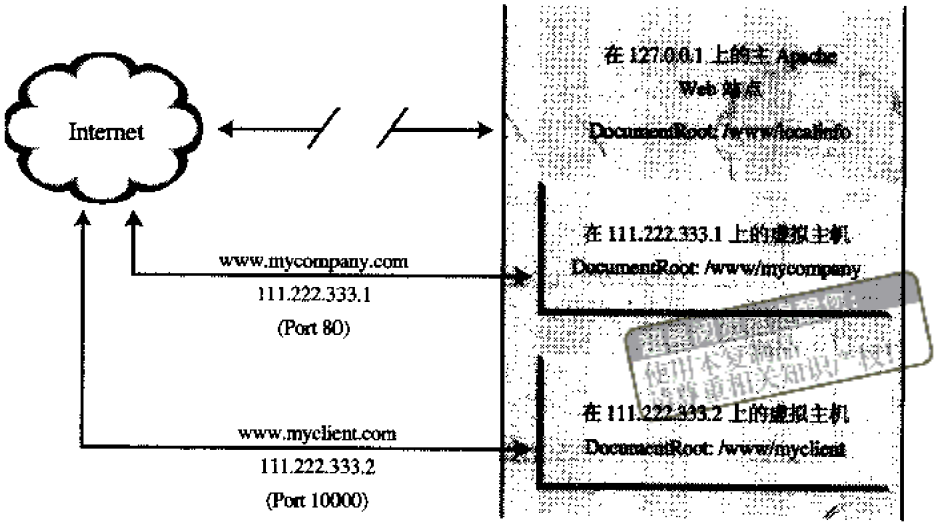

图 6-11 在不同端口上的基于 IP 虚拟主机的例子

```
...
</VirtualHost>

<VirtualHost 111.222.333.2>
DocumentRoot /www/myclient
ServerName www.myclient.com
...
</VirtualHost>
```

正如前面的例子一样，主服务器只能捕捉到来自本地用户的请求。www.mycompany.com 服务器能通过标准的端口（有效的 URL 是 www.mycompany.com/和 www.mycompany.com:80/）进行访问。然而，www.myclient.com 站点只能通过端口 10000 进行访问。因此要访问这个站点就需要使用以下的 URL：

```
http://www.mycompany.com:10000/
```

现在让我们看一个基于名称的实例配置。

6.5.4 基于名称的虚拟主机

这个例子说明如何配置基于单个 IP 地址的主机，webserver.mycompany.com，来建立两个基于名称的虚拟 Web 站点：www.mycompany.com 和 www.myclient.com。图 6-12 表示这个例子所讨论内容的简化图。正如所看到的，这个 Web 服务器机器只有一个单独的 IP 地址，它具有如下的一些 DNS 记录：

```
www.mywebserver.com    IN    A       111.222.333.1
www.mycompany.com      IN    CNAME   webserver.mycompany.com.
www.myclient.com       IN    CNAME   webserver.mycompany.com.
```

www.mycompany.com 和 www.myclient.com 是 webserver.mycompany.com 的两个别名（CNAME 记录）。

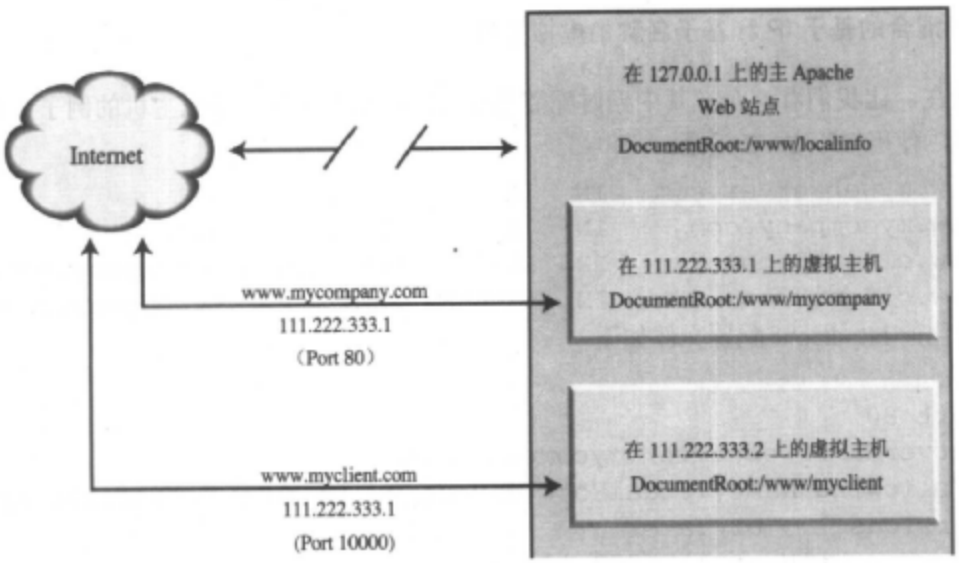

图 6-12　基于名称的虚拟主机的例子

简化的 httpd.conf 配置文件如下：

```
...
Port 80
ServerName webserver.mycompany.com

#Following NameVirtualHost directive is only needed for Apache
versions 1.3 or later
NameVirtualHost 111.222.333.1

#first (or primary) virtual server
<VirtualHost 111.222.333.1>
DocumentRoot  /www/mycompany
ServerName  www.mycompany.com

...
</VirtualHost>

<VirtualHost 111.222.333.1>
DocumentRoot  /www/myclient
ServerName  www.myclient.com
...
</VirtualHost>
```

主服务器只服务于系统的本地用户。使用非 HTTP/1.1 兼容的浏览器的用户能看到第一台虚拟主机。除此之外，所有使用 HTTP/1.1 兼容的浏览器的用户都能请求正确的虚拟主机，并且这台服务器按照在配置文件中找到的相应的虚拟主机设置为它们提供服务。现在，让我们看一个混合的（基于 IP 和基于名称的）虚拟服务器例子。

6.5.5 混合的基于 IP 和基于名称的虚拟主机

现在，让我们看一个在其中同时配置基于 IP 和基于名称的虚拟主机的例子。在这个例子中，使用以下 DNS 记录。

```
www.mywebserver.com   IN   A       111.222.333.1
www.mycompany.com     IN   A       111.222.333.2
www.client1.com       IN   CNAME   webserver.mycompany.com.
www.client2.com       IN   CNAME   webserver.mycompany.com.
```

简化的 httpd.conf 配置文件如下：

```
...
Port 80
ServerName webserver.mycompany.com
#Following NameVirtualHost directive is only needed for Apache
versions 1.3 or later
NameVirtualHost 111.222.333.1

<VirtualHost 111.222.333.1>
DocumentRoot /www/client1
ServerName www.client1.com
...
</VirtualHost>
<VirtualHost 111.222.333.1>
DocumentRoot /www/client2
ServerName www.client2.com
...
</VirtualHost>

<VirtualHost 111.222.333.2>
DocumentRoot /www/mycompany
ServerName www.mycompany.com
...
</VirtualHost>
```

当组合基于 IP 和基于名称的虚拟主机时，记住，在所有使用同一个 IP 地址的基于名称的虚拟主机前使用 NameVirtualHost 命令。如果想保持配置文件可读，那么只需要每个 IP 地址键入一条 NameVirtualHost 命令，并在它的后面列出所有使用该 IP 地址的基于名称的虚拟主机。

6.5.6 使用_default_的虚拟主机

这个例子表明如何捕获每一个到任意非指定 IP 地址和端口（即没有被任何一台虚拟主机使用的地址/端口的组合）的请求。

简化的 httpd.conf 配置文件如下：

```
...
<VirtualHost _default_:*>
```

```
DocumentRoot /www/default
...
</VirtualHost>
```

使用此类带有通配符端口的缺省虚拟主机能有效地阻止任何上述的请求进入主服务器。缺省的虚拟主机从不服务于送往基于名称的虚拟主机使用的地址/端口组合的请求。如果一个请求含有未知（的地址/端口）或没有 Host:头标，它通常由基于名称虚拟主机提供服务（在配置文件中对应特定地址/端口组合的、最先出现的基于名称的虚拟主机）。

然而，如果你想为不同端口地址提供不同的缺省虚拟主机，也可以做得到。看下面的例子。

简化的 httpd.conf 配置文件如下：

```
...
<VirtualHost _default_:80>
DocumentRoot /www/default80
...
</VirtualHost>

<VirtualHost _default_:*>
DocumentRoot /www/default
...
</VirtualHost>
```

用于端口 80 的缺省虚拟主机（它必须出现在任何带有通配符端口的缺省虚拟主机的前面）捕获所有被发送到未指定的 IP 地址的请求。主服务器从不用于为任何请求服务。如果只想为端口 80 提供缺省的虚拟主机，那么就不要提供通配符端口虚拟主机。

6.5.7　将基于名称的虚拟主机转变为基于 IP 的虚拟主机

在这个例子中，假定对一个基于名称的虚拟主机 www.myclient1.com 分配了 IP 地址 111.222.333.3。为了在 Apache 配置文件中反映这些变化，需要为 www.myclient1.com 修改这个虚拟主机设置，如下：

```
...
...
NameVirtualHost 111.222.333.1

<VirtualHost  111.222.333.1  111.222.333.3>
DocumentRoot  /www/client1
ServerName www.myclient1.com
...
</VirtualHost>
```

注意，VirtualHost 命令现在有旧的 IP 地址 111.222.333.1 和新的 IP 地址 111.222.333.3。这样做以便对这台虚拟主机的基于名称和基于 IP 的请求都能得到服务。立即删除旧的 IP 的作法是不可取的，因为许多域名服务器和代理服务器可能已捕获旧的 IP，它们可能无法找到这个站点，直到其修改了 DNS 数据。这个虚拟主机 www.myclient.com 现在通过新地址（作为基于 IP 的虚拟主机）和旧地址（作为基于名称的虚拟主机）都能进行访问。

6.6 建立虚拟主机的限制因素

现在让我们看一下影响服务器具有的大量虚拟主机的各种因素。UNIX 操作系统限制了一个进程可以使用的文件描述符（也称为文件 handle）数量。一般限制在 64 个，并且可能被提高到大的硬限制（hard-limit）。当使用大量虚拟主机时，Apache 可能会用尽可用的文件描述符。

Apache 为每个打开的日志文件使用一个文件描述符，并且在内部用去大概有 10~20 个附加的文件描述符。如果 Apache 使用文件描述符接近系统设置的限制，它可能试图为自己分配更多的文件描述符。然而，如果系统不提供 setrlimit()系统调用、如果这个调用在系统上不能工作、如果所需的描述符数超出了系统的最大允许值，或者如果系统有其他影响文件描述符个数的限制，这种努力可能失败。

如果遇到这种情况，可以尝试两种选择来解决它：

♦减少 Apache 使用的日志文件数。可能需要检查 VirtualHost 条目，并删除每个虚拟主机的记录日志的功能。

♦在许多 UNIX 系统上，可以尝试在运行 Apache 之前手工设置这个文件描述符限制。像下面这样使用 ulimit 命令：

```
ulimit -S -n 100
```

将限制数设为 100 个描述符。根据你的需要，试试不同的数目。要了解详细情况，请参阅 ulimit 手册。

到此为止，已讨论了 Apache 虚拟主机配置的各个方面。它使你能够建立自己的虚拟 Web 站点。现在将话题转到虚拟主机的另一个方面。它和 Apache 没有直接联系，但对于拥有许多虚拟 Web 站点的系统来说仍然很重要。

6.7 用于虚拟 Web 站点的 sendmail 配置

正如前面提到的，虚拟 Web 站点对服务商是经济的，对用户也一样。代替在 Internet 上使用专用的服务器系统，一个机构（或个人）能够在 ISP 上拥有虚拟 Web 站点并使它充分发挥作用。虚拟 Web 站点确实是专业 Internet 展示的必要条件。为了使它更具有真实感，许多机构还希望他们的 e-mail 地址能反映其虚拟 Internet 域名。例如，如果需要猜想一个 Microsoft 程序员的 e-mail 地址的话，那么很可能会猜想它以@microsoft.com 结束，而不是以@aol.com 结束，对吗？从那些拥有 Internet 访问权的任何现代公司的角度看，也期望如此。所以，迫切需要对虚拟 Web 站点和虚拟 e-mail 服务器两者同时提供支持的虚拟 Internet 域。下面讨论如何改动配置以建立虚拟 SMTP e-mail 服务器。

因为 sendmail 是最流行的（搞不懂为什么对它配置仍如此令人困惑？）SMTP 邮件服务器，在这一节中用它作为例子。

为简单起见，假定需要为 nitec.com 的虚拟 Internet 域 milkyweb.com 设置虚拟 SMTP

邮件服务器。还假定 nitec.com 的邮件服务器是 mail.nitec.com，它被用作 milkyweb.com 的虚拟邮件服务器。

6.7.1　对虚拟 SMTP 邮件服务器的 DNS 配置

超星阅览器提醒您：

在配置邮件服务器（sendmail）之前，必须确保对已有的邮件服务进行正确的 DNS 设置。记录邮件到达位置的 DNS 记录称作 MX 记录。

在这一虚拟域的 DNS 数据库文件（milkyweb.db）中，键入如下一行：

```
IN MX 10 mail.nitec.com.
```

这行正好在 NS 记录行的后面。它表示用于 milkyweb.com 的 SMTP 邮件服务器是 mail.nitec.com，并将它的优先权设置为 10。如果 nitec.com 有多个邮件服务器，并且它们都被配置为接收 milkyweb.com 的邮件，那么你可以添加不同优先权和邮件主机的多个 MX 记录。例如，如果在 nitec.com 中有另外一台称为 postoffice.nitec.com 的邮件主机，它能接收 milkyweb.com 的邮件，可以添加如下：

```
IN MX 10 mail.nitec.com.
IN MX 20 postoffice.nitec.com.
```

注意这些优先权数是 10 和 20。它们是任意的，但有一条规则：数越小，优先权越高。因此，对于前面的例子，如果 Internet 邮件服务器需要发送邮件给 milkyweb.com，它将首先试用 mail.nitec.com 主机来传递邮件。如果出于某种原因 mail.nitec.com 不可用，那么它将试用 postoffice.nitec.com 服务器。

通过使用 nslookup 实用程序，能确保正确配置 MX 记录。在 nslookup 提示符下键入下列命令就可以完成这件事：

```
Set q=mx
milkyweb.com
```

一旦完成 DNS 配置，设置虚拟邮件支持的下一步是编辑 mail.nitec.com 系统的 /etc/sendmail.cw 文件。sendmail 配置在各版本之间可能存在差异。尽管如此，可以事先查询有关文档。它们可以在如下的 URL 上找到：

```
www.sendmail.org/
```

下列配置已在 sendmail: 8.8.5/8.8.5 进行了检验。

6.7.2　配置/etc/sendmail.cw

/etc/sendmail.cw 文件包含 sendmail 守护进程服务的域名。在这种情况下，在这个文件中添加 milkyweb.com 域：

```
#sendmail.cw - include all aliases for your machine here.
nitec.com
milkyweb.com
```

下一步将为实际用户转换表建立一个虚拟用户。