

邱仲潘 钟锒光
陈 宏 李立峰 编著

Linux

系统 集成



北京航空航天大学出版社

<http://www.buaapress.cn.net>

Linux 系统 集成

邱仲潘 钟铨光 编 著
陈 宏 李立峰

北京航空航天大学出版社
<http://www.buaapress.cn.net>

内 容 简 介

本书是介绍 Linux 系统集成的实用指南。内容包括对数据库服务器、网页服务器、代理服务器、邮件服务器等的介绍,可以完全通过自由软件建立一个真正实用的系统,书中特别是对目前很流行而又缺乏中文文档的 PHP 进行了深入介绍。任何希望利用 Linux 进行系统集成的读者,可从本书中学到一些实用的知识,使自己的系统更坚实、更巧妙。

图书在版编目(CIP)数据

Linux 系统集成/邱仲潘编著. —北京:北京航空航天大学出版社,2000.2

ISBN 7-81012-958-9

I. L... II. 邱... III. 操作系统, Linux-基本知识
IV. TP316.89

中国版本图书馆 CIP 数据核字(1999)第 55549 号

Linux 系统集成

邱仲潘 钟铨光 编 著
陈 宏 李立峰

责任编辑 马广云

责任校对 李宝田

北京航空航天大学出版社出版发行

北京市学院路 37 号(100083) 发行部电话:(010)82317024 发行部传真:(010)82328026

<http://www.buaapress.cn.net>

E-mail:pressell@publica.bj.cninfo.net

河北省涿州市新华印刷厂印刷 各地书店经销

*

开本:787×1092 1/16 印张:15.5 字数:397 千字

2000 年 7 月第 1 版 2000 年 7 月第 1 次印刷 印数:5000 册

ISBN 7-81012-958-9/TP·385 定价:24.00 元

前 言

1998 年 Linux 在中国真正掀起了热潮。一年来, Linux 的推广工作进行得如火如荼, Linux 用户数量猛增。Turbo Linux、XTeam Linux 和红旗 Linux 等中文版操作系统也相继推出, Linux 书籍更是如潮涌来。但是, Linux 是工具, 而不是玩具。如何让丰富的 Linux 资源真正为企业系统服务? 许多系统集成人员都尝试着在这方面能有所作为, 并且也取得了一定的成就。为了让更多开发人员少走弯路, 更有效地利用 Linux 资源进行系统集成, 本书作者系统地总结了这方面的工作, 编著了此书。

很多人都在 Hotmail 站点中申请了免费邮件账户。这个站点的邮件服务器正是本书介绍的 Qmail 服务器。Yahoo 邮件、XOOM.com 和帽子 Linux 邮件列表等大型网站都是使用 Qmail 做邮件服务器。

现有的局域网多是 Novell 网和 NT 网。在 Linux 上用户可以用 Mars 仿真 Novell 文件服务, 用 SAMBA 仿真 NT 文件服务。而且可免费实现这一功能, 并用完全公开源代码的系统, 用户可以根据自己的需要随意定制。

微软公司的 ASP 取得了巨大的成功, 但它只适用于微软公司的平台。PHP 的功能同样强大, 而且是跨平台的, 在网上更有生命力, 而且免费并公开源代码。不少网络爱好者在 BBS 上询问哪里有 PHP 中文文档介绍? 本书即可满足网络爱好者们这一需求。

读者可能已看到很多价格以万计的大型数据库, 但可能没有发现在 Linux 世界里也有强大而免费的数据库系统。本书将着重介绍 Mysql 和 Postgres。读者可以发现, 从商业软件中走出来, 自由软件的世界也很精彩。

书中还有很多内容, 读者可以慢慢领略。为了降低本书成本, 作者尽量精简图形, 以便在有限的篇幅内向读者提供最丰富的信息。

本书第一~五章由钟铨光编写, 第六和第七章由陈宏博士执笔, 李立峰编写了第八章。本书由邱仲潘选题、策划、统筹, 并最后统稿、审定。由于经验不足, 加上时间仓促, 本书难免有错漏之处, 欢迎广大读者不吝指教。

作 者

1999 年 10 月

目 录

第一章 绚丽的 Apache 世界

1.1 简介	1
1.2 功能特性	2
1.3 安装与升级	2
1.3.1 下载	2
1.3.2 编译	2
1.3.3 编译常见问题	3
1.3.4 运行配置	4
1.3.5 运行 Apache	6
1.3.6 效果测试	6
1.3.7 常见运行问题	6
1.4 高级专题	8
1.4.1 虚拟主机	8
1.4.2 分布式服务	11
1.4.3 存取控制与用户认证	11
1.5 常见问题	17
1.6 Apache 命令一览	19
1.6.1 核心命令列表	19
1.6.2 Apache 标准模块	29
1.6.3 Apache 附带模块	44

第二章 PHP

2.1 简介	49
2.2 技术支持	52
2.3 编译与安装	52
2.3.1 准备工作	52
2.3.2 下载源码	53
2.3.3 编译设置	53
2.3.4 编译与安装	56
2.3.5 测试和性能	57
2.4 配置 php3.ini	57
2.4.1 语言选项	57
2.4.2 数据处理命令	58
2.4.3 路径和目录命令	59
2.4.4 资源控制命令	59

2.4.5 错误处理命令	60
2.4.6 调试配置命令	60
2.4.7 扩展装入命令	60
2.4.8 邮件配置命令	61
2.4.9 数据库配置命令	61
2.4.10 其他配置命令	61
2.5 安 全	61
2.5.1 CGI 模式可能的攻击	62
2.5.2 作为 Apache 模块时的安全问题	63
2.6 PHP 变量与表达式	63
2.6.1 PHP 标记	63
2.6.2 变量类型	64
2.6.3 变量作用域	66
2.6.4 变名变量	68
2.6.5 外部变量	68
2.6.6 环境变量	69
2.6.7 类型转换	69
2.6.8 运算符	70
2.6.9 表达式	71
2.7 PHP 控制流语句	72
2.7.1 IF	72
2.7.2 IF ...ELSE...	72
2.7.3 ELSEIF	73
2.7.4 IF(): ... ENDIF	73
2.7.5 WHILE	74
2.7.6 DO...WHILE	74
2.7.7 FOR	74
2.7.8 SWITCH	75
2.7.9 REQUIRE	76
2.7.10 INCLUDE	76
2.8 函数及 PHP 常用函数	76
2.8.1 PHP 函数初步	76
2.8.2 变量类型函数	77
2.8.3 数组函数	78
2.8.4 字符串函数	80
2.8.5 时间函数	82
2.8.6 程序执行函数	84
2.8.7 目录函数	84
2.8.8 文件系统函数	85

2.8.9 网络函数	89
2.8.10 邮件函数	90
2.8.11 数学函数	90
2.8.12 输出函数	92
2.8.13 HTML 函数	92
2.8.14 HTTP 相关函数	93
2.8.15 URL 函数	94
2.8.16 session 处理函数	95
2.8.17 杂项函数	97
2.9 PHP 和数据库	98
2.9.1 一个简单的 Mysql 数据库连接范例	98
2.9.2 数据库函数中的 Mysql 部分	100
2.9.3 封装 ODBC 操作	104
2.10 PHP4——升腾的新星	108
2.10.1 简介	108
2.10.2 新特性	108
2.10.3 不兼容性	110
2.10.4 从 PHP3 升级	110
2.10.5 PHP4 的效率	111
2.10.6 技术支持与问题讨论	112

第三章 Qmail 快捷通信与安全沟通

3.1 简介	113
3.2 Qmail 邮件服务基本概念	114
3.3 安装 Qmail	114
3.3.1 安装要求	114
3.3.2 下载源文件	115
3.3.3 创建 Qmail 工作目录	115
3.3.4 创建运行 Qmail 的组和用户	115
3.3.5 编译 Qmail	115
3.3.6 配置	115
3.3.7 设置常用 Qmail 别名	116
3.3.8 信件投递	117
3.3.9 测试本地投递	118
3.3.10 删除 Sendmail	118
3.3.11 设置 Qmail SMTP 服务器	119
3.3.12 设置 POP3 服务器	119
3.4 安装 daemontools 与 tcpserver 扩展 Qmail	119
3.5 垃圾邮件	121
3.6 活用 Dot - Qmail	121

3.7	虚拟主机	122
3.8	使用拨号的企业邮件服务器	123
3.9	Webmail	124
第四章 Mysql		
4.1	简介	128
4.2	特性	128
4.3	Mysql 的兼容性	129
4.4	Mysql 的稳定性	130
4.5	2000 年问题	131
4.6	寻求帮助和报告问题	132
4.7	许可和版权问题	133
4.8	安 装	133
4.8.1	系统要求	133
4.8.2	获取软件	134
4.8.3	安 装	134
4.8.4	初始化及测试	141
4.8.5	性能测试	143
4.9	自动启动和中止 Mysql	144
4.10	使用初步	145
4.10.1	连接和断开服务器	145
4.10.2	使用 SQL 语句	146
4.11	Mysql 的存取权限控制	146
4.12	Mysql 与 C/S 结构的 MIS 系统开发	148
4.12.1	ODBC 支持	148
4.12.2	Delphi 开发支持	149
4.12.3	在 Win32 平台上使用 SSH 连接	149
4.13	Mysql 语法	149
4.13.1	字 符	149
4.13.2	数 字	150
4.13.3	数据库、表、索引、字段和别名的命名	150
4.13.4	字段类型	151
4.13.5	字段索引	152
4.13.6	运 算	153
4.14	Mysql 函数	153
4.14.1	比较函数	153
4.14.2	控制流函数	154
4.14.3	数学函数	154
4.14.4	字符串函数	155
4.14.5	日期时间函数	157

4.14.6 杂项函数	160
4.14.7 分组函数	161
4.15 数据库操作语句	162
第五章 Postgres 数据库	
5.1 简介	172
5.2 下载	172
5.3 安装	173
5.3.1 安装前的准备	173
5.3.2 安 装	173
5.4 初始化	174
5.5 Postgres 初探	176
5.5.1 连接服务器	176
5.5.2 创建数据库	177
5.5.3 存取数据库	178
5.5.4 删除数据库	178
5.5.5 Postgres SQL 特性	178
5.6 Linux 平台上的常见问题	180
5.6.1 编译阶段相关问题	180
5.6.2 附属程序编译问题	182
5.6.3 运行阶段相关问题	182
5.7 配置 Win95 客户端的 ODBC 驱动	184
5.8 在 Windows NT 上安装 Postgres 服务器	185
5.8.1 安装 Cygwin B20.1	185
5.8.2 安装 Cygwin B20 的开发工具	185
5.8.3 安装 Ludovic Lange 的 Cygwin32 IPC Package	186
5.8.4 安装 PostgreSQL 6.5.2 源代码	186
5.9 应用示例	187
第六章 Mars	
6.1 简介、发展与现状	189
6.1.1 局域网简介	189
6.1.2 Novell 仿真器简介	189
6.1.3 Mars 简介	190
6.2 相关网址	190
6.3 Mars 程序包	190
6.4 安装与升级	191
6.4.1 安装要求	191
6.4.2 编译设置	191
6.4.3 编 译	192
6.4.4 配置 nw.ini 的选项并安装	193

6.5	配置 Mars	193
6.6	应用范例	206
6.7	常见问题及技巧	207
第七章 Samba		
7.1	简介、发展与现状	209
7.2	相关网址	210
7.3	Samba 文件层次	210
7.4	安装/升级	210
7.5	配 置	211
7.5.1	全局设定	211
7.5.2	设定共享卷	213
7.5.3	配置打印机	213
7.5.4	查看 Samba 服务器当前连接状态	213
7.5.5	全局浏览	214
7.5.6	修改口令	214
7.5.7	初步测试	215
7.6	应用范例	215
7.7	常见问题及技巧	217
7.7.1	Samba 服务程序的启动/关闭	217
7.7.2	跨网关浏览的方法	217
第八章 IP 伪装、Socks5 与 Squid		
8.1	IP 伪装	218
8.1.1	简 介	218
8.1.2	IP 伪装原理	218
8.1.3	安 装	219
8.2	Socks5 代理服务	221
8.2.1	简介及背景	221
8.2.2	基本原理	221
8.2.3	Linux 下的 Socks5 安装和基本配置	222
8.2.4	配置 Socks5 服务详解	224
8.2.5	多界面及用户认证的 Socks5 代理设置	226
8.2.6	技术支持	226
8.2.7	客户端说明	226
8.3	Squid 代理	229
8.3.1	简介及背景	229
8.3.2	基本原理	229
8.3.3	下载、编译和安装	230
8.3.4	基本配置及运行	230
8.3.5	高级配置	231

第一章 绚丽的 Apache 世界

1.1 简介

在缤纷美丽的 WWW 世界中, Web 服务器是个默默无闻的幕后英雄。它根据客户浏览器的要求提供给客户相应的文档。用户只看到浏览器处理后的漂亮页面。Internet 上的 Web 站点不计其数, 是什么软件在后面支撑着这么多的服务器呢?

Web 服务器种类极其繁多。例如, 在 Windows NT/95 平台上采用的 Web 服务器有微软 IIS (Internet 信息服务器)、PWS (个人 Web 服务器); 而在 Unix/Linux 平台上, 则以 Apache 最为流行。诸如 www.yahoo.com 和 www.deja.com 这样的超大型站点都采用 Apache 作为 Web 服务器。

Apache 从出现起就表现了势不可挡的发展势头, 很快就成为最流行的 Web 服务器。到 2000 年 5 月, 根据 Netcraft 对 1 500 万台服务器的统计调查(<http://www.netcraft.co.uk/survey/reports>), Internet 上 60% 的 Web 服务器采用了 Apache。如果包括 Apache 的衍生产品, 这个比例会更高, 比其他 Web 服务器的总和还要多, 是微软 IIS 份额(21%)的 3 倍多。对域名以 net 结尾的服务器统计结果更加惊人, Apache 份额达到 75%。

Apache 不仅功能强大, 而且还是免费的自由软件, 甚至可以在商业用途中免费使用 Apache。

说到 Apache 的名字, Apache 组织对其解释是“A PAtCHy Server”, 因为 Apache 是基于 NCSA Web 服务器并对其加以改进后写成的。

Apache 是采用标准 C 语言写成的, 并且开放源代码; 所以, 它能用于多种桌面服务器操作系统, 如 Unix 和 Windows NT。不过就性能和稳定性来说, Apache 的 Win32 版本还不如 Unix 平台的版本。

虽然 Apache 是免费软件, 但是它支持最新的 HTTP 标准, 而且 Apache 开发小组的工作热情相当高。目前, Apache 的版本已经开发到了 1.3.12, 推出新版本的速度非常快。而采用新技术的 Apache2.0 目前也进入了 Alpha 测试阶段。另外, Apache 还有一些衍版本, 例如 - Apache 和 Java - Apache。

Apache 不存在 2000 年问题。Apache 从来不用 2 位数记年。为了和旧程序兼容, Apache 也可以识别 ANSI C 的 `asctime()` 函数和 RFC850 - /RFC1036 - 风格的时间格式。函数 `asctime()` 使用 4 位年号, 而 RFC850 和 RFC1036 使用 2 位年号, 这时 Apache 把 70 以下的值认为是 20xx 年。

大多数的 Unix 使用一个无符号的 32 位整数计时, 其值为从 1970 年 1 月 1 日开始的秒数。所以, 2038 年更可能出问题, 不过新版本的 Unix 系统一般都解决了这个问题。

1.2 功能特性

Apache 与商业软件相比,其功能只多不少。而且 Apache 非常容易配置,可以完全按用户需要定制编译。它有如下特性:

- 非常方便地替换原来的 NCSA Web 服务器,改正了 NCSA 1.3 和 1.4 的错误,速度比 NCSA 服务器大大加快。
- 符合 HTTP/1.1 规范。
- 支持虚拟服务器。例如, <http://www.a.com> 和 <http://www.b.com> 其实只是定位到一台服务器的不同目录而已。
- 自定义错误处理。
- Apache 支持多种用户认证方式。还可以使用第 3 方模块来支持更多当前流行的网络认证系统。
- Apache 服务器性能非常好。它采用了“进程预生成”模式。一般服务器只在接到请求后才生成子进程。服务完成后,子进程就死亡了。下一次再重复这个循环。在 Unix 下面,创建一个子进程代价很昂贵。而 Apache 是预先创建一定数量的子进程,这些子进程一直重复为新的请求服务。这样减少了创建、清除子进程的系统消耗。如果连接要求多于预先生成的子进程,Apache 则会根据需要动态地改变其数目。HTTP 1.1 规范中的 Keep-Alive 固定连接允许多个客户请求使用一个 TCP 连接。对有多张图片的页面,性能可提高 50%。不过要求预先知道页面的大小,所以,对动态生成的页面无效。
- Apache 不提供加密传输;但是,基于 Apache 的 Apache-SSL 和 Apache-SSL-US 支持加密传输,它们都兼容 Netscape 的加密机制。
- Apache 的源代码是公开的,可以在某些特殊的 Unix 平台上自行编译。如果对软件的安全性要求很高,则可以自行研究源代码。
- 在新闻组 comp.infosystems.www.servers 中有非常多的 Unix 高手,可以在那里提出问题。推荐使用 <http://www.deja.com> 网站来搜索新闻组(newsgroup),它是个非常好的新闻组数据库。可以很快地在那里找到想要的信息。
- Apache 采用模块化设计,可以开发自己的模块来扩展其功能。

1.3 安装与升级

1.3.1 下载

在 <http://www.Apache.org> 网页上,可以看到最新的有关 Apache 发行版本和测试版本的消息;可以下载源代码自己编译,也可以下载预先编译的可执行文件。这里介绍的是下载源代码并在 Linux 上安装编译。下载 Apache 的网址是 <http://www.Apache.org/dist/>。

1.3.2 编译

编译分成 3 步:首先选择需要使用的模块,然后生成编译配置文件,最后进行编译。

1. 选择模块

选择模块就是在模块文件中注释或打开相应的模块,或者添加自己编写或下载的新模块。Apache 发行包里附带的模块都经过了广泛的测试和应用,性能和安全性上都是值得信赖的。也可以在 <http://www.Apache.org/dist/contrib/modules/> 中找到其他各种各样的模块,不过它们的安全性和性能就不一定有那么可靠了。

2. 编译设置

在 Apache 目录中输入以下命令:

```
/Apache/src% ./Configure
```

显示如下:

```
Using 'Configuration' as config file
```

```
+ configured for <whatever> platform
```

```
+ setting C compiler to <whatever> *
```

```
+ setting C compiler optimization-level to <whatever> *
```

如果想生成多种配置,则在 Configure 后面加一个参数。例如:

```
/Apache/src% ./Configure -file 配置文件名
```

如果有特殊要求或者编译失败了,则必须编辑配置文件的 EXTRA_CFLAGS、LIBS、LFLAGS、INCLUDES。成功运行 Configure 后,会生成一个 makefile 文件。

3. 编译

输入

```
make
```

```
make install
```

成功编译后,在 src 目录中应该有一个 httpd 可执行文件。在可执行发行包里直接就能找到这个文件。

Apache 还有一些支持程序,位于发行目录的 support 子目录里。进入相应目录,键入 make 即可编译支持程序。

1.3.3 编译常见问题

1. gcc 报告编译错误

Apache 要求 gcc 的编译版本和操作系统的版本一致。如果升级了操作系统,则必须重新编译 gcc;否则,Apache 将不能通过编译。可以试用 gcc-v 来查看 gcc 是在哪个版本下编译的。

2. Redhat Linux 5.0 报告 crypt() 函数有问题

Redhat 是基于 Glibc 的,它把 crypt() 函数放在单独的库里。编辑 src/Configure 文件,设置 EXTRA_LIBS=-lcrypt,然后重新运行 src/Configure。

3. 在 Linux 下提示“shmget: function not found”

因为编译核心时没有包括 SysV IPC 支持,请重新编译核心。

4. 得到错误提示“__inet_ntoa”或其他的“__inet_*”没有定义

如果系统安装了 BIND-8,那么头文件和库通常会发生冲突。BIND-8 的文件安装在 /usr/local/include/ 和 /usr/local/lib/ 中,而系统文件通常安装在 /usr/include/ 和 /usr/lib/ 中。如果系统使用 /usr/local/include 中的头文件和旧库,那么将发生冲突。所以,必须确保要么都使用老版本,要么都使用新版本。在配置文件的 EXTRA_LDLAS 中加入 -lbind,然后重新

编译配置。

5. 在 Linux 下编译, 警告 struct iovec 重复定义

C 库的头文件和核心头文件发生冲突。请确认它们是否匹配。可以从 C 库的头文件 `/usr/include/sys/uio.h` 中删除 `struct iovec` 定义, 或者编译设置中在 `EXTRA_CFLAGS` 行使用 `-DNO_WRITEV`。这会影响性能, 除非实在没有办法, 否则不要使用。

1.3.4 运行配置

Apache 在 `conf` 目录下有 3 个配置文件 `access.conf`、`httpd.conf` 和 `srn.conf`。不过从版本 1.3.6 开始, Apache 建议只使用 `httpd.conf`。安装后该目录含有相应的示例文件, 分别命名为 `access.conf-dist`、`httpd.conf-dist` 和 `srn.conf-dist`。

```
cd conf
cp httpd.conf-dist httpd.conf
cp access.conf-dist access.conf
cp srm.conf-dist srm.conf
```

有了配置文件, 现在就可以着手配置了。Apache 的配置文件由很多命令节组成, 命令节由符号“`<... >... </... >`”形式的标签分隔, 其内部则是一些具体的配置命令。例如:

```
<Directory somedir/in/your/tree>
Directive option option
Directive option option
</Directive>
```

Apache 的配置命令非常多, 而且每个新模块都会带来很多新命令。本章后面会专门介绍 Apache 的配置命令, 不过也不可能非常完整。每次推出 Apache 新版本, 都会带来一些新命令。下面将重点介绍一些常用的配置命令。

1. ServerAdmin

参数是 Web 管理员的 Email 地址。Web 服务器出现问题时, 用户用该 Email 地址和管理员联系。

2. User and Group

User 和 Group 为服务器的 UID 和 GID, 一般是使用缺省值 `nobody` 和 `nogroup`。可以用 UID 和 GID 的数字形式。注意查看在 `/etc/passwd` 中是否有 `nobody` 用户, 在 `/etc/group` 中是否有 `nogroup` 组。对于指定的 UID, ID 必须只拥有很低的权限, 这是因为 CGI 程序通常会以同样的权限运行。如果 Apache 以 `root` 或者其他用户运行, 且这个 CGI 又有安全漏洞, 则很可能会被人恶意利用。

如果启动时报告“`setgid: Invalid argument`”, 请查看配置文件中的 Group 命令参数是否存在于 `/etc/group` 文件中。例如:

```
User nobody
Group nogroup
或者用如下数字形式:
User #-1
Group #-1
```

3. ServerName

ServerName 为服务器运行的域名, 会返回给客户。如果需要在台单机上测试 Apache, 则可以设置 ServerName 为 localhost。这样就可以用 `http://www.localhost` 来浏览了。

4. ServerRoot

ServerRoot 用于指定 Web 服务器 httpd 文件所在目录的绝对路径。

5. DocumentRoot

Documentroot 用于指定服务器文档的根目录绝对路径。类似的还有 UserDir。UserDir 命令为每个用户定义一个个人主页目录, 不过这也可能带来安全问题。一个用户可能创建一个其他目录的连接, 并将其放在个人主页目录中发布。这样任何人都可以浏览整个文件系统。

这里设置的存取权限可以被目录里面的 .htaccess 文件设置取代。当然也可以设置忽略所有的 .htaccess 设置, 这只需要使用命令 `AllowOverride none` 即可。AllowOverride 的缺省值是 all。

6. ServerType

通常 Apache 使用 standalone 模式或者 inetd 模式, 这可以用 ServerType 命令来设置。采用 ServerType inetd 可以把服务器设置成 inetd 模式。standalone 模式的性能比 inetd 模式的好。使用一旦启动, 总是有一个进程在处理客户的请求。如果是用 inetd 模式, 则每次接收到新请求后, 都会启动一个新服务器进程。这样带来的资源消耗特别大。所有 ServerType 的缺省值是 standalone。如果服务器负载非常低, 也可以使用 inetd 模式。inetd 模式的好处就是适合测试配置, 因为这样每次 HTTP 请求时, 服务器都重新启动一个新进程。该进程使用最新配置。这样改动的配置马上就能生效。如果使用 standalone 模式, 则不得不重新启动服务器。

inetd 意思就是 internet superserver, 在服务器启动时通过 /etc/rc 启动。一旦启动, inetd 就等待连接。一旦发现有连接要求, 它就启动相应的程序来为连接服务。请求结束后则自动结束该程序。要在 inetd 中启动 Apache, 需要编辑 /etc/inetd.conf 和 /etc/services 两个文件。

首先编辑 /etc/services 文件, 这个文件的每一行对应一项服务。一行中包含服务名、端口、协议名和服务别名 4 项。每项用 Tab 或空格分隔。通常该文件已经包含了以下一行:

```
http 80/tcp httpd httpd
```

其中 80 是 HTTP 的缺省端口。也可以指定为一个 0~65 535 之间的整数。不过保留的端口号只有超级用户才能使用。所以, 通常使用 inetd 模式一般运行在 8080 端口, 即

```
http 8080/tcp httpd httpd
```

另外, 还需要编辑 /etc/inetd.conf 来配置 inetd 以等待 HTTP 请求。每一行包括服务名、socket 类型、协议、是否等待、程序运行的身份、服务器程序和程序参数。例如:

```
httpd stream tcp nowait nobody/usr/local/etc/httpd/httpd httpd -f/usr/local/etc/httpd/conf/httpd.conf
```

这一行的意思是服务器以 nobody 身份运行。通常 standalone 的服务器是用 root 用户启动的, 这样才能绑定到 80 端口上; 然后可以用 User 和 Group 命令来改变其子进程的 UID 和 GID。但是在 inetd 模式下, 如果指定为 root 用户, 则每个进程都是 root 的, 这样会有安全隐患。在 inetd 中加入 httpd 后, 就可以重新启动 inetd 了。先用 ps 命令找到 inetd 的进程号, 然后使用

```
# kill -HUP InetdProcessID
```

1.3.5 运行 Apache

Apache 的主程序名缺省是 httpd。该文件有很多参数,例如:

```
http [-d serverroot][ -f configurationfile][ -x][ -v][ -?]
```

各参数说明如下:

- -d 设置 serverroot 目录,可以取代配置文件的设置。
- -f 指定配置文件的目录。如果是相对目录格式,则是相对 serverroot 目录而言的。缺省是 serverroot/conf/httpd.conf。
- -x 用于调试,一般用户不需要使用。它不会生成子程序。
- -v 输出 Apache 的版本。
- -? 输出使用帮助。

运行 Apache 有两种模式,即 standalone 模式(独立模式)和 inetd 模式。

如果是独立服务器模式,则可以用手动运行 httpd。例如:

```
/usr/local/etc/httpd/httpd -f /usr/local/etc/httpd/conf/httpd.conf
```

如果 Apache 配置成使用 80 号端口,则必须用 root 身份启动 httpd;否则,只能使用 1 015 以上的端口。

如何把 Apache 设置成开机自动启动呢?如果配置成 inetd 服务器模式,则不需要启动 inetd 服务器,inetd 会自动发出相应请求来启动它。如果是 standalone 模式,则在/etc/rc.local 文件中加上

```
# Run httpd server automatically
echo 'starting httpd server'
if [ -f /etc/rc.httpd ]; then
sh /etc/rc.httpd
fi
```

然后在 /etc 目录中创建一个名为 rc.httpd 的文件,内容如下:

```
#!/bin/sh -u
cd /usr/local/etc/httpd/
./httpd -d /usr/local/etc/httpd/ -f conf/httpd.conf
```

当然,也可以把这几行直接加到 rc.local 里;不过,上面的方法中将 rc.httpd 名字更改后就可以方便地关闭 httpd 服务了。

1.3.6 效果测试

配置完成后,看一看效果(图 1.1),马上就能体会到 Apache 世界的魅力。在浏览器中输入 URL http://localhost,就可以看到 Apache 是否有运行。现在会看到 documentroot 目录下的 index.html 文件。一般情况下都不会有什么问题。如果有错,则 Apache 都会告诉错误原因,以方便问题的解决。

1.3.7 常见运行问题

1. Apache 找不到配置文件

错误提示如下:

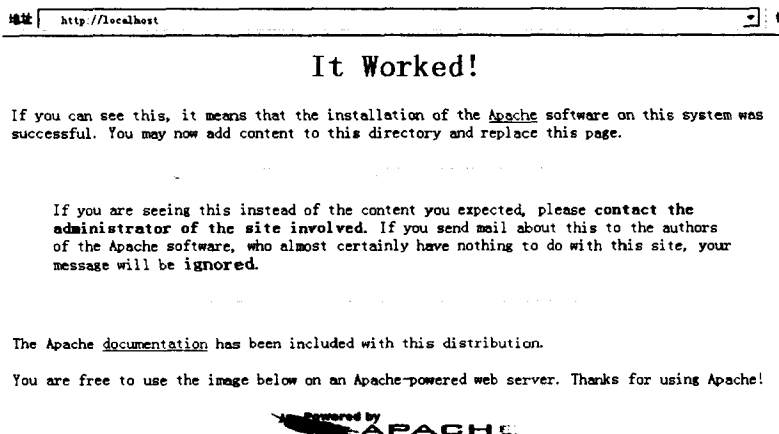


图 1.1 Apache 正常工作界面

```
httpd:could not open document config file:/usr/local/etc/httpd/conf/httpd.conf
fopen:No such file or directory
```

即 Apache 找不到配置文件。这种情况一般是因为在编译 Apache 时指定了非缺省的 server-root, 而配置文件不在该目录中。

2. 端口发生冲突

错误提示如下:

```
httpd:could not bind to port
bind:Address already in use
```

这可能是因为服务器已经运行了其他 Web 服务器, 端口发生冲突了; 或者是非 root 用户试图使用 80 号端口。

3. 服务器没有反应

查看 httpd 有没有运行。正常情况下使用 `ps ax | grep httpd` 应该看到多个 httpd 在运行。

4. Apache 崩溃

这种情况极其少见。如果经常碰到, 则最好更新 Apache 版本。

5. Apache 挂起, 或者不能启动

错误记录文件中有“fcntl:F_SETLKW:No record locks available”或类似错误。这是个与锁定有关的问题, 通常意味着服务器试图使用一个 NFS 文件系统上的 synchronization 同步文件。在并行操作模式下, Apache 服务器在存取一些资源时需要提供一些同步操作。而同步操作需要锁定文件。如果系统不支持, 则会导致上述错误。NFS 文件系统正好不支持这种方法。如果需要在 NFS 上面使用, 则应在配置文件中加上一行 `LockFile/var/run/Apache-lock`。这个路径必须是可写的, 不要用 `/var/tmp`。

6. htm 文件不能按照 html 格式显示

查找到 mime.types 文件, 查看里面是否有一行关于 htm 文件的配置。如果没有, 则加上, 然后重新启动 Apache。

7. 其他错误

查看 logs/error.log 和 logs/access.log 这两个文件。它们记载了客户请求和 Apache 错误。