

Linux 服务平台构建 与 JSP 编程实战

孙洪雁 李玉环 主编



科学出版社
www.sciencep.com

1 00
T
TP316.89

92

Linux 服务平台构建与 JSP 编程实战

孙洪雁 李玉环 主编

科学出版社

北 京

内 容 简 介

JSP 是一种全新的、跨平台的网页编程技术, 具有独特而强大的功能, 尤其是其跨平台操作的软件复用性, 可大大减轻开发人员的工作量, 缩短系统的开发周期, 因而, 在商业交互系统的开发领域具有很强的竞争力和良好的发展前景。

本书是作者多年来建立和维护网站的经验总结, 是一本非常实用的网站编程指南教材, 书中所给出的实例 JSP 代码和后台 Java 程序代码均采用了最新 Java 类库, 所有代码均在实际运行调试中通过, 可以直接引用, 快速上手。对编程过程中可能出现的典型错误进行了程序运行演示, 并给出了解决方案。通过在教学过程中不断补充、修改和完善, 最终形成现在的知识架构。本书系统完整, 叙述由浅入深, 符合读者认知规律, 既可作为本科、高职高专的教材使用, 又可供自学及应用参考。

图书在版编目(CIP)数据

Linux 服务平台构建与 JSP 编程实战/孙洪雁, 李玉环主编.—北京: 科学出版社, 2006

ISBN 7-03-016792-9

I. L … II. ①孙… ②李… III. ①Linux 操作系统②JAVA 语言—程序设计 IV. ①TP316.89②TP312

中国版本图书馆 CIP 数据核字 (2006) 第 003452 号

责任编辑: 吕建忠 赵卫江/责任校对: 都 岚

责任印制: 吕春珉/封面制作: 飞天创意

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京彩色印装有限公司 印刷

科学出版社发行 各地新华书店经销

*

2006 年 3 月第 一 版 开本: 787×1092 1/16

2006 年 3 月第一次印刷 印张: 26 1/4

印数: 1—3 000 字数: 608 000

定价: 35.00 元

(如有印装质量问题, 我社负责调换〈路通〉)

销售部电话 010-62136131 编辑部电话 010-62138017 (H101)

本书编写人员名单

主 编 孙洪雁（山东科技大学）

李玉环（山东农业大学）

参 编 孙红梅（山东科技大学）

谭建辉（广东阳江职业技术学院）

朱小平（广东科学技术职业学院）

李义勇（山东农业大学）

贾瑞生（山东科技大学）

李风燕（山东农业大学）

张 琮（解放军后勤工程学院）

张东升（山东科技大学）

任 远（东南大学）

丁 义（北京邮电大学）

主 审 任永功 梁永全（分内容审稿）

前 言

JSP (Java server page) 是一种全新的、跨平台的网页编程技术, 它不仅具有自己独特而强大的功能, 而且解决了 ASP、PHP 各自独成一家、平台间不兼容或处于弱势的第三方中间件运行等问题, 虽然许多网页编程人员习惯了 ASP 或 PHP, 但是 JSP 的发展有其必然性, 这不仅是因为它具有独特而强大的功能, 更重要的是它能跨平台操作的软件复用性, 这大大减轻了开发人员的工作量, 极大地缩短了系统的开发周期, 它的软件复用特性在商业开发领域更具有无比的竞争力, 这必将推动交互式动态网站系统的快速发展。

与 ASP 和 PHP 一样, JSP 编程对于一个普通的网页设计者来说, 是比较容易掌握的, 它不像 C、Perl 等, 这些编程语言需要程序设计者掌握大量的编程技巧, 并要具有比较丰富的编程经验, 其中的正则表达式、函数、类等编程要件复杂而抽象, 编程环境的构建、程序的试验与调试也费时费力, 因而, 生成一个分布式应用系统具有一定的难度。JSP 则不同, 它使用 Java 组件作为后台运行程序, 而 Java 是面向对象的编程技术, 其系统内建类封装了大量的应用需求, 省略了程序员大量的基础性劳动, 不像 C 或 Perl, 每一个函数都要由程序员自己去编写。由此可见, JSP 为网站及网页设计者提供了理想的工具, 与 ASP 和 PHP 相比, 它的跨平台操作性能是最大优势, 平台的无关性可以使设计者有更多的精力从事系统的编程工作, 而不必因为平台问题而耽误过多的时间。

虽然 JSP 具有众多的独特的优势, 特别是跨平台操作的良好性能, 但这并不意味着平台问题就无关紧要, 笔者在多年的网络管理及编程工作中发现, 有时操作系统平台对系统编制工作的影响并不亚于编程工具本身所存在问题的影响。现在, 有关网页编程的工具书很多, 在 Internet 上也能够查阅到大量的相关资料, 但这些书籍及材料大多是在某一个技术上的突破性研究或应用技巧, 或者是英文资料的编译, 而对使用这一技术需要具备的条件和所需环境的技术细节着墨不多, 加之环境因素变化更新较快, 因而很难找到具体的解决方案, 至于根据原始材料所产生的经过实际环境运行的实例更是难得一见, 偶尔有这样的资料, 但也难以得到错误信息的相关解释, 这往往使程序设计者和环境构建者感到十分无奈, 这样的问题相信大多数的网页编程者和网络管理员都或多或少地遇到过, 而且所耗费的精力也相当大, 有时会无奈地放弃一个很好的编程工具, 甚至重新构建新的操作系统平台, 从而使新的编程技术无法得到有效的发挥。如何顺利构建这样一个环境, 是一个新技术应用的基础, 也是这个技术被用户接受和进一步发展的基础。所以平台与编程工具的结合问题是本书讨论的重点之一。本书将从 JSP 开发者的角度, 用一定的篇幅来介绍如何快速构建一个可靠而实用的平台, 而不去深入研究操作平台内部机制的更多技术细节。

Linux 是一种基于 UNIX 操作系统内核的源代码开放的操作系统, 它为世界各地的编程爱好者提供了操作系统开发的公共平台和应用程序编制平台, 由于世界各地众多编程者的参与, 从而使 Linux 迅速发展成为当今一个功能强大的操作系统, 并逐渐成为 21 世纪操作系统领域的热门系统, 形成了对微软 Windows 系统强有力的竞争态势。由于 Linux 正处于

不断发展壮大和普及阶段,应用程序开发商为了能够在不久的将来立足于该领域,都不遗余力地推出自己的 Linux 免费应用软件,以尽快获得用户认可,为今后的商业发布打下基础, Linux 这些特点必将在今后一个时期形成一个良性竞争局面,从而使用户最终受益。因此,使用 Linux 可以获得大量的免费软件和低价软件,这就可使用户以较小的软件投资迅速构建出一个强大的网络操作平台或桌面应用系统。

用于网络平台的 Linux 是一个分布式操作系统,能够进行网络节点的架设、网络管理、应用程序的开发等,具有十分广泛的应用前景,与其他分布式操作系统相比,除价格优势外, Linux 更加轻便实用,安全配置更加灵活,系统运行稳定可靠,笔者所使用的另一台 RedHat Linux 8.0 服务器主机(本书在 RedHat Linux 9.0 上写作),集 DNS、HTTP、SendMail、vsFTP、MySQL 等应用服务器,以及 SSH、OpenWebMail 等应用服务程序及开发工具于一身,虽然在启用后的两年时间里增加了不少的应用配置,却未曾重启过一次机器,由此可见,它的稳定性已远远超过了非源码开放系统,这一点在世界上其他大公司对 Linux 的应用过程中也同样得到了证明。2002 年 9 月 4 日, IBM 新西兰公司和新西兰航空公司同时宣布, IBM 将帮助新西兰航空将其企业服务器全面转向 Linux, 合同金额 3700 万美元。该公司之所以摒弃其他系统而转向 Linux, 主要就是看中了 Linux 的运行稳定性。同时, 由于它的系统配置和应用程序运行环境配置均提供了文本操作方式, 因而, 更容易被用户理解和发现攻击行为, 它的内核可编译功能, 使用户进行开发时更加得心应手。所有这些, 都是 Linux 越来越受用户欢迎的重要原因。

笔者认为, JSP 发布者虽然最终摆脱了平台限制, 但其函数等编程要件的开发最初还是从 UNIX 平台上开始的, 这就决定了 JSP-Linux 的先天适配性, 加上 Linux 独特的网络支持优势, 二者应为最佳的搭配。虽然 Linux 在网络架设方面具有诸多独特的优势, 而在桌面应用上, 它还算不上是一个主流的操作系统, 因此, 大多数用户对其认识程度还不如 Windows 那样深刻, 许多人也知道 Linux 的诸多优点, 但往往因为没有多少亲身经历, 或是仅掌握了一些支离破碎的知识而感到无从下手。根据这种情况, 本书分为三大部分, 第一部分为 Linux 网络操作平台的搭建; 第二部分为使用 JSP 所必需的知识与 JSP 开发环境及应用支持环境的构建; 第三部分为 JSP 开发。通过阅读本书, 读者可以系统地掌握从平台搭建到 JSP 开发的全过程, 从而避免因为一些无关的过程而浪费宝贵的开发时间, 以达到快速构建平台和快速开发最终产品的目的。当然, 本书所提供的 Java 及 JSP 代码也完全可以用于 Windows 平台应用开发。

本书有以下重要特点: 书中所编写的实例程序均采用了 Java 官方网站发布的最新类库, 所有代码均在实际运行调试中通过, 可以直接引用, 快速上手; 本书是作者多年来建立和维护网站的经验总结, 是一本非常实用的网站编程指南教材, 即使在 Windows 下编程, 其提供的大量准确代码仍然可以被引用; 对编程过程中可能出现的典型错误进行了程序运行演示, 并给出了解决方案, 这一点是大多数作者容易忽视而读者又十分需要的; 在教学过程中不断补充、修改和完善, 最终形成现在的知识架构。本书系统完整, 叙述由浅入深, 符合读者认知规律, 既可以作为本科、高职高专的教材使用, 又可供自学与应用参考。

本书由孙洪雁拟订章节大纲, 全书的统稿、定稿由孙洪雁、李玉环共同完成。张东升、任远、丁义主要参与第 1 章的撰写, 李风燕、张琮主要参与第 2 章的撰写, 朱小平、贾瑞生主要参与第 3 章的撰写, 孙红梅、谭建辉主要参与第 4 章的撰写, 孙洪雁、李玉环完成

第 5、6、7 章的撰写，第 8 章介绍的用户注册管理系统是一个完整的网络应用软件系统实例，系统的需求分析、设计和代码的编写，由山东农业大学网络与教育技术部李义勇完成，并将全部源程序代码奉献给读者，这些代码的移植与调试工作由贾昊林完成。写作过程中使用了山东省果树研究所提供的 Linux 网络平台，为此，对该所领导一并表示感谢。

本书由广东省科学技术职业学院任永功研究员主审，山东科技大学博士生导师梁永全教授分内容审稿；在此深致谢意！由于计算机技术发展迅速，成稿时间仓促，书中错误与不当之处在所难免，希望广大读者批评指正。

作 者

2005 年 11 月

目 录

第 1 章 Linux 基础知识	1
1.1 Linux 概述	1
1.2 Linux 基本类型及其构建理论	2
1.3 Linux 分布式结构	5
习题	5
第 2 章 Linux 服务器系统构建	6
2.1 IP 地址与域名准备	6
2.1.1 IP 的获取和域名的申请	6
2.1.2 用预留 IP 定义网络	10
2.2 上网连接方式	11
2.2.1 连接方式	11
2.2.2 硬件的选型	14
2.3 服务器系统的布置	18
2.3.1 硬盘准备与程序安装	18
2.3.2 DNS 配置	23
2.3.3 在用户终端上远程操作服务器	29
2.3.4 配置二级域名、WWW、FTP、Telnet 和数据库	31
2.3.5 电子邮件配置	45
2.3.6 Qmail 布署	52
2.3.7 基于 Qmail 的完整 Webmail 解决方案	58
2.3.8 安全的服务器策略	64
习题	72
第 3 章 SQL 语言	73
3.1 数据定义与数据控制	73
3.1.1 SQL 的数据定义	73
3.1.2 SQL 的数据控制	75
3.2 数据查询语句	76
3.2.1 SELECT 语句的一般格式	76
3.2.2 SELECT 语句的几种特殊格式	78
3.2.3 筛选数据记录的 WHERE 语句	80
3.3 数据操作语句及操作数	83
3.3.1 DELETE 语句: 删除数据	83
3.3.2 UPDATE 语句: 更新数据	83
3.3.3 INSERT INTO 语句: 增加数据	83

3.3.4	SELECT INTO 语句: 建立新数据表.....	84
3.3.5	操作数.....	85
3.3.6	ALL,DISTINCT,DISTINCTROW,TOP 属性词用法.....	86
3.3.7	PARAMETERS (参数) 声明的用法.....	87
3.3.8	多层 SQL 查询.....	87
3.4	条件子句应用范例.....	87
3.4.1	HAVING 条件子句.....	87
3.4.2	IN 条件子句.....	88
3.4.3	FROM 条件子句.....	88
3.4.4	WHERE 条件子句.....	89
3.5	BETWEEN...AND 运算符.....	89
3.6	SQL 常用函数.....	89
3.7	SQL 语法总结分类.....	90
	习题.....	91
第 4 章	Java 语言基础.....	93
4.1	基本概念.....	93
4.1.1	标识符与关键字.....	93
4.1.2	基本数据类型.....	94
4.1.3	变量与常量.....	95
4.2	数组与流程控制.....	96
4.2.1	数组.....	96
4.2.2	流程控制.....	99
4.3	类与继承.....	103
4.3.1	类.....	103
4.3.2	类的继承.....	113
4.4	接口和异常.....	116
4.4.1	包.....	116
4.4.2	接口的介绍.....	118
4.4.3	异常.....	121
	习题.....	123
第 5 章	JSP 平台构建.....	126
5.1	网站开发平台概论.....	126
5.1.1	常用功能.....	126
5.1.2	交互功能的应用.....	127
5.2	与 JSP 平台构成有关的几个重要概念.....	128
5.2.1	JDK.....	129
5.2.2	Tomcat 与 Tomcat+Apache.....	129
5.3	JSP 平台的构建过程.....	133
5.3.1	程序部署.....	133

5.3.2 JSP 平台构建中的错误处理与 DSO 支持	138
5.4 mod_jk2.SO 共享模式整合	144
习题	151
第 6 章 JSP 网站编程基础	152
6.1 JSP 基础知识	152
6.1.1 JSP 与 Web 的概述	152
6.1.2 JSP 技术特点	154
6.1.3 一个简单 JSP 运行实例	158
6.2 JSP 基本指令应用	160
习题	174
第 7 章 简单的 JSP 程序设计	175
7.1 简单的 JSP 网页程序设计	175
7.1.1 简单 JSP 网页的创建与运行	175
7.1.2 程序分析	177
7.2 JSP 与 Java Servlet	179
7.2.1 编写第一个 Servlet	180
7.2.2 表单实例	190
7.3 JSP 与 JavaBeans	201
7.3.1 编写第一个 JavaBeans	202
7.3.2 JavaBeans 的方法与 JSP 的两个属性标签	207
7.4 JSP 与 Applets	212
7.4.1 第一个 Applets	212
7.4.2 Applets 对象操作	217
7.4.3 关于与 Java 异常有关的操作	226
7.4.4 关于 Java 中多线程操作	227
7.4.5 Applets 新方法的使用	231
习题	244
第 8 章 JSP 应用实例编程	246
8.1 访问计数器程序的制作	246
8.1.1 文件构成与设计思路	246
8.1.2 文字模式的访问计数器	247
8.1.3 用 JavaBeans 模块化文字模式的访问计数器	254
8.1.4 制作图形模式的访问计数器	258
8.2 网上投票系统的编程	264
8.2.1 系统文件构成与程序流程图	265
8.2.2 系统编程	266
8.3 留言板程序的制作	282
8.3.1 系统文件构成	282
8.3.2 表单的编写	283

8.3.3 保存留言	294
8.3.4 留言的删除与发送	308
8.4 Java Mail 编程	321
8.4.1 Java Mail API 的基本概念	321
8.4.2 建立 Java Mail 应用环境所需要的软件	323
8.5 用 JSP 操作 MySQL 数据库	335
8.5.1 在 JSP 中使用 JDBC 来连接 MySQL	335
8.5.2 用数据库 Beans 直接赋值	335
8.5.3 用 JavaBeans 增加、修改、删除记录	337
8.6 一个完整的用户注册阅读管理系统	339
8.6.1 数据库的规划	341
8.6.2 建立连接数据库的类	344
8.6.3 编写注册程序	345
8.6.4 编写用户登录程序	346
8.6.5 编写管理程序	346
8.6.6 编写用于系统调用的 JSP 网页	348
习题	387
附录 1 JSP 数据库连接方式	393
附录 2 jar 命令的使用	398
附录 3 建立 JSWDK 调试环境	399
附录 4 Linux 下 MySQL 的安装与交互操作	402
主要参考文献	408

第 1 章 Linux 基础知识

1.1 Linux 概述

Linux 是一种基于 UNIX 操作系统内核的源代码开放的操作系统，可以在 PC 机上运行，于 1991 年由芬兰赫尔辛基大学的学生 Linus Torvalds 开发。Linus Torvalds 将 Linux 的源程序在 Internet 上公开后，世界各地的编程爱好者自发组织起来对 Linux 进行改进，并编写了许多应用程序，从而使今天的 Linux 迅速发展成为一个功能强大的操作系统，并逐渐成为 21 世纪操作系统领域的热门系统，形成了对微软 Windows 系统强有力的竞争态势。

Linux 的开发及它的源代码受 GNU (general public license, 一般公共许可) 保护，它对每个人都是完全免费的，GNU 又恰是 GNU's Not UNIX 的缩写，此句被当初不满 UNIX 实施版权控制，而继续倡导自由软件思想的 UNIX 系统开发者，也就是 Linux 的创始人用以表明 Linux 与 UNIX 是完全不同的系统，其区别不仅是 Linux 是倡导自由软件思想的产物，更重要的是 Linux 的奋斗目标是要发展成为一个比 UNIX 更加强大的操作系统，这一点在当前操作系统领域已逐步得到体现。基于 GNU，人们可以开发具有个性化的 Linux 操作系统，这样的操作系统又是受版权保护的，这就意味着一个完整的 Linux 发行版在 GNU 控制范围内，仍然可以在网上免费获取，但根据 GNU 修正案，版权人可以以其他媒体（如磁盘和光盘等）向公众出售，这一方面是对版权的鼓励，另一方面解决了不能从网上获取系统的用户需求，从而加速了系统的推广速度。不过，与实行版权控制的系统相比，其销售价格是十分低廉的。同样，在 Linux 下开发的一些应用系统软件也是如此，但针对某个用户的应用开发则不受这一协议的限制。

使用 Linux 的用户在软件投入上会感到十分轻松，只需要很少的投资便可以建立起一个完整而安全的操作系统平台或网络操作平台。这主要是由于两方面的原因：一是 Linux 完整的操作系统可以由多家公司开发，也可以由多家软件和硬件公司合作按照统一的源代码开放标准制定统一的通信标准来共同开发适于某个区域的网络通信平台，这就使不同系统之间存在着不可避免的竞争，同时，Linux 正处于蓬勃发展的阶段，以优惠用户的方式和横向联合的方式迅速获取足够的市场份额也是商业竞争的重要手段之一；二是由于 GNU 省略了大量的编程劳动，缩短了一个完整系统的开发周期，从而使开发者的开发成本大大降低。此外，由于 Linux 正处于前景十分看好的普及阶段，应用程序开发者为了能够在不久的将来立足于该领域，都不遗余力地推出自己的 Linux 免费应用软件，以尽快获得用户认可，为今后的商业发布打下基础。

Linux 既可以用于桌面系统，又可以用于网络平台，用于网络平台的 Linux 是一个分布式操作系统，能够进行网络节点的架设、网络管理、应用程序的开发等，具有十分广泛的用途，与其他分布式操作系统相比，Linux 更加轻便实用，安全配置更加灵活，

同时, 由于它的系统配置和应用程序运行环境配置均提供了文本操作方式, 因而, 更容易被用户理解和发现攻击行为, 笔者曾以 RedHat Linux 8.0 做过 4 年实际验证, 无需额外防火墙, 只要充分调动系统固有的安全资源进行合理规划, 其安全防护效果令人十分满意, 到 2004 年 9 月, 仅 SSH 恶意登录每天大约在 20 次以上, 还未见成功入侵者, 由此可见, 与其他昂贵操作系统相比, Linux 是一种性价比十分可观的操作系统, 它正以其高性能、低价位获得越来越广泛的用户支持。1997 年, Linux 的一个发行包 RedHat 5.0 被评为最佳操作系统, 从而真正拉开了 Linux 开始角逐分布式操作系统和桌面系统市场的序幕。

企鹅是 Linux 的吉祥物, 它是由 Linux Torvalds 挑选的代表他所创立的 Linux 操作系统。

1.2 Linux 基本类型及其构建理论

可从两个方面对 Linux 进行分类, 一是按版权许可分类, 二是按内核分类。按版权许可分类即按公司发布版进行分类, 这是用户选择系统的重要依据, 这类系统有基本 Linux、Magic Linux、RedHat Linux、Slackware Linux、FreeBSD 及红旗 Linux 等系统。其中, 基本 Linux 系统是完全免费的, 而公司在此基础上开发的操作系统, 则是根据 GNU 修正案进行回收成本或说低盈利商业发布的系统。按内核分类是 Linux 主要的分类方法, 这种分类方法是商业版发布商生产用于不同目的或不同市场需求的 Linux 的主要依据, 同时也是应用编程者所必须掌握的。

无论如何分类, 对于 Linux 来说, 其系统的构建都离不开分布式系统所应遵循的协议。各种不同的 Linux, 其解决问题的方法都有其共同的特性。解决复杂性的所有方法都基于一个基本原理, 即问题分解和各个击破。也就是说, 都是把大型的、难以解决的问题(或系统)分解成一定数量的复杂度较低的子问题(或子系统), 再根据需要重复这一过程直到每一部分都小到可以解决为止, 而各种方法只是这种原理的一些不同运用而已。

下面介绍一下 Linux 系统所涉及的一些概念。

1. 层次

将解决方案分解成若干部分, 在这些部分中存在一个问题域的最底层, 它为上层的抽象层次(layer)较高的工作提供基础。较高层建立在其低层基础之上。OSI 和 TCP/IP 协议堆栈是众所周知的层次化软件设计的成功例子。操作系统设计的层次化解决方案可能会包含一个可以直接和硬件通信的层次, 然后在其上为更高层提供抽象支持的层次。这样更高层就可以对磁盘、网卡等硬件进行访问, 而并不需要了解这些设备的具体细节。

层次化设计的一个特征是要逐步构建符号集(vocabulary)。随着层次的升高, 符号集的功能将越来越强大。层次化设计的另外一个特征是完全可以在对其上下层透明的条件下替换某一层。在最理想的情况下, 移植层次化的操作系统只需要重写最底层的代码。纯层次化模型实现的执行速度可能会很慢, 因为高层必须(间接的)通过调用一系列连续的低层才能处理完自己的任务, 即 N 层调用 $N-1$ 层, $N-1$ 层调用 $N-2$ 层等,

直到实际的工作在 0 层被处理完成。接着，结果当然是通过同样的路径反向传递回来。因此，层次化设计通常会包含对某些高层直接通信和某些低层通信的支持；这样虽然提高了速度，但是却使得各个层次的替换工作更加困难（因为不止一个高层会直接依赖于这个人们所希望进行替换的层次）。

2. 模块

模块（module）将具体的一部分功能块隐藏在抽象的接口背后。模块的最大特点是将接口和其实现分离开来，这样就能够保证一个模块可以在不影响其他模块的情况下进行改变。这样也将模块之间的依赖关系仅仅限定于接口。模块的范围是试图反映求解域内一些方面的自然的概念性界限。纯模块化的操作系统因而就可能有一个磁盘子系统模块、一个内存管理子系统模块等。纯模块化和纯层次化的操作系统之间的主要区别是，一个可以由其他模块自由调用，模块间没有上层和下层的概念（从这个意义上来说，模块是广义的层次。按照纯粹的观点，层次是最多可供一个其他模块调用的模块，这个模块也就是它的直接上层模块）。

3. 对象

对象（object）和模块不同，因为对于初学者来说它们具有不同的考虑问题方式，实现的方法也可能各自独立。但是，就当前的目的来说，对象不过是结构化使用模块的方法。组件（component）作为对象思想的进一步改进，目前还没有在操作系统设计中广泛使用。即使如此，也没有足够的理由将其和模块划分在不同的范畴中。

一种观点是强调内核视图是层次化的，而且是体系结构无关层次位于体系结构相关层次之上（更为精确的视图是在顶层增加一个附加的体系结构相关的层次。这是因为系统调用接口位于应用程序和内核之间，而且是体系结构相关的）。而另一种观点则着重强调更加模块化的内核视图。从合理的表述层次上看，这两种观点都是正确的。但也可以说这两种观点都是错误的。事实上，内核是遵从所有人们所能够指出的设计原则集合的，因为它就是从众多思想中抽取出来的。这就是说，Linux 内核既不是严格层次化的，也不是严格模块化的，更不是严格意义上的任何类型，而是以实用为主要依据的（实际上，如果要用一个词来概括 Linux 从设计到实现的所有特点，那么实用就是最确切的）。也许最保守的观点是内核的实现是模块化的，虽然这些模块有时会为了追求速度而有意跨越模块的界限。

这样，Linux 的设计同时兼顾了理论和实际。Linux 并没有忽视设计方法；相反，在 Linux 的开发基本思想中，设计方法的作用就像是编译器，是一个完成工作的有力工具。选择一个基本的设计原则（如对象）并完全使用这种原则，不允许有任何例外，这对于测试该原则的限制，或者构建以说明这些方法为目的的教学系统来说都是一个不错的方法。但是如果要用它来达到 Linux 的设计目标则会引起许多问题。而且 Linux 的设计目标中也并不包括要使内核成为一个完全纯化的系统。实际上，如果对于 Linux 来说是正确的，那么它们对于所有最成功的设计来说都是正确的。最成功、应用最广泛的实际系统必然是实用的系统。有些开发人员试图寻找功能强大的可以解决所有问题的特殊方法。他们一旦找到了这种方法，所有的问题就都迎刃而解了。像 Linux 内核一样的成功设计通常需要为系统的不同部分和描述上的不同层次使用不同的方法。这样做的结果可能不是很清晰，也不是很

纯粹，但是这种混合产物比同等功能的纯粹系统要强大而且优秀得多。

Linux 大部分都是单内核的操作系统，内核可能是微内核，也可能是单内核（后者有时称之为宏内核，即 `macrokernel`）。按照类似封装的形式，这些术语定义如下。

4. 微内核

在微内核（`microkernel`）中，大部分内核都作为独立的进程在特权状态下运行，它们通过消息传递进行通信。在典型情况下，每个概念模块都有一个进程。因此，如果在设计中有一个系统调用模块，那么就必然有一个相应的进程来接收系统调用，并和能够执行系统调用的其他进程（或模块）通信以完成所需任务。

在这些设计中，微内核部分经常只不过是一个消息转发站：当系统调用模块要给文件系统模块发送消息时，消息直接通过内核转发。这种方式有助于实现模块间的隔离（某些时候，模块也可以直接给其他模块传递消息）。在一些微内核的设计中，更多的功能（如 I/O 等）也都被封装在内核中了。但是最根本的思想还是要保持微内核尽量小，这样只需要把微内核本身进行移植就可以完成将整个内核移植到新的平台上。其他模块都只依赖于微内核或其他模块，并不直接依赖硬件。

微内核设计的一个优点是在不影响系统其他部分的情况下，用更高效的实现代替现有文件系统模块将会更加容易，甚至可以在系统运行时将开发出的新系统模块或者需要替换现有模块的模块直接而迅速地加入系统。另外一个优点是不需要的模块将不会被加载到内存中，因此，微内核就可以更有效地利用内存。

5. 单内核

单内核（`monolithic kernel`）是一个很大的进程，它的内部又可以被分为若干模块（或者是层次或其他）。但是在运行时，它是一个独立的二进制大映像。其模块间的通信是通过直接调用其他模块中的函数实现的，而不是消息传递。

单内核的支持者声称微内核的消息传递开销引起了效率的损失。微内核的支持者则认为因此而增加的内核设计的灵活性和可维护性可以弥补任何损失。

这些问题在这里不做重点来讨论，不过，这种争论经常会令人想到前几年 CPU 领域中 RISC 和 CISC 的斗争。现代成功的 CPU 设计中包含了所有这两种技术，就像 Linux 内核是微内核和单内核的混合产物一样。Linux 内核基本上是单一的，但是它并不是一个纯粹的集成内核。内核模块系统将微内核的许多优点引入到 Linux 的单内核设计中。

为什么 Linux 必然是单内核的呢？一个方面是历史的原因：在 Linux 的观点看来，通过把内核以单一的方式进行组织并在最初的空间中运行是相当容易的事情。这种决策避免了有关消息传递体系结构、计算模块装载方式等相关工作（内核模块系统在随后的几年中又进行了不断改进）。

另外一个原因是充足的开发时间的结果。Linux 既没有开发时间的限制，也没有来自于市场压力的发行进度。所有的限制只是并不过分的对内核的修改与扩充。内核的单一设计在内部实现了充分的模块化，在这种条件下的修改或增加并不困难。而且问题还在于没有必要为了追求尚未证实的可维护性的微小增长而重写 Linux 的内核（Linux 曾多次特别强调了如下的观点：为了这点利益而损耗速度是不值得的）。后面章节中将详

细讨论充足开发时间的效果。

如果 Linux 是纯微内核设计,那么向其他体系结构上的移植将会比较容易。实际上,有一些微内核,如 Mach 微内核,就已经成功地证明了这种可移植性的优点。实际的情况是, Linux 内核的移植虽然不是很简单,但却是可能的,向一个全新的体系结构上的典型的移植工作需要 30000~60000 行代码,再加上不到 20000 行的驱动程序代码(并不是所有的移植都需要新的驱动程序代码)。粗略计算一下,一个典型的移植大约平均需要 50000 行代码。这对于一个程序员或者一个程序小组来说是可以办到的,在一年之内就可完成。虽然这比微内核的移植需要更多的代码,但是 Linux 的支持者将会提出,这样的 Linux 内核移植版本比微内核更能够有效地利用底层硬件,因而移植过程中的额外工作是能够从系统性能的提高上得到补偿的。

这种特殊设计的权衡也不是很轻松就可以决定的,单内核的实现策略公然违背了传统的看法,后者认为微内核是未来发展的趋势。但是由于单一模式(大部分情况下)在 Linux 中运行状态良好,虽然内核移植相对来说比较困难,但没有明显地阻碍程序员们的工作,他们已经成功地把内核移植到了现存的大部分实际系统中,更不用说类似掌上型电脑了。只要 Linux 的众多特点仍然值得移植,新的移植版本就会不断涌现。

1.3 Linux 分布式结构

大体上说, Linux 仍然还算是一个分布式的层次结构,但是由于它的发展一直走着一条实用的路线,众多的参与者使它不再完全拘泥于 ISO 层次模型的限制,从而使层次变得模糊起来,这是追求系统效率和放宽对第三方软件限制的必然结果,但无论如何,从其结构的本质上来说,结构的层次还是存在的,如图 1-1 所示。

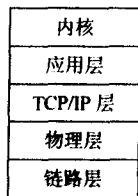


图 1-1 Linux 的体系结构

按层次模型的要求,对于 TCP/IP 层的请求,应由该层提交给应用层,由应用层将问题分解后,交给内核处理,处理的结果再按相反的路径返回 TCP/IP 层。但事实上, Linux 内核可以直接处理来自于 TCP/IP 层甚至是物理层的事务,这就使应用层和 TCP/IP 层的界限变得模糊了。但是,并不是所有的参与者都不遵循这一体系结构,只是可以不去遵循它而已。

习 题

1. 简述 Linux 的起源与发展。
2. Linux 与 Windows (网络操作系统)有什么根本的不同?它们的相同点是什么?
3. Linux 与 UNIX 在版权控制上有什么不同? Linux 遵守什么样的版权协议?该协议对 Linux 的发展有什么意义?
4. 与严格意义上的分布式操作系统相比, Linux 在层次模型上有什么特点?
5. 什么叫内核?与 Windows 相比, Linux 内核有什么特点?

第2章 Linux 服务器系统构建

2.1 IP 地址与域名准备

租用异地服务器空间组建信息中心虽然可以大量地节省投资,但对于建设一个较为完备的信息中心来说,这并不是一个好办法,因为没有硬件资源的管理权限,很难将工作做得比较完美。

软、硬件环境的准备包括三个方面:一是 IP 资源的获取和域名申请;二是内部网络设备的购进与连接;三是出口的租赁。

2.1.1 IP 的获取和域名的申请

1. IP 地址的注册

IP (Internet protocol) 是一个网际协议,根据这个协议进行编码,这个编码就是 IP 地址,把一个 IP 地址分配给一个在网上工作的网络设备(如计算机、交换机、路由器等),这个设备在网上就有了一个确定的身份,由于 IP 地址是唯一的,因此,在 Internet 上可定位这个设备。可以在 IP 经销商处进行注册,如果不打算使用过多的 IP,则可以就近从分销商那里获取。如中国教育和科研计算机网网络信息中心(CERNIC)就是一个全国范围的 Internet 资源注册管理部门,负责全网的 IP 地址分配、域名注册。注册前应从经销商那里获取申请表和相关协议,填写并加盖法人公章后交经销商确认,确认通过后方可获取 IP 的使用权。有关 IP 的注册信息可在各经销商的官方网站上获取。

2. 域名的注册

域名(domain name)是网站在互联网上的名字。独立的域名是一个网站的第一笔无形资产,它随网站访问量的增加而不断增值。域名应简单、易记并具有特色。与 IP 一样,在 Internet 上,域名也是唯一的,不同的是,域名是由网站拥有者自己命名的。因此,建站前应首先在域名经销商那里注册域名。

(1) 域名的含义 在网络传输技术中,域名是一个在 Internet 中用于解决地址对应问题的一种方法。是一个技术名词。但随着 Internet 在全球的普及,域名也就带上了浓厚的人文和社会科学的色彩。因而,域名已成为了 Internet 文化的重要组成部分。

在商界,域名已被誉为“企业的网上商标”。产品商标是企业的生命,而域名巨大的商业价值已被全世界的企业所认识。1998 年 3 月的一个月,世界上注册了 179331 个通用顶级域名,平均每天注册 5977 个域名,每分钟 25 个!这个记录正在以每月 7% 的速度增长。中国国内域名注册的数量,从 1996 年底之前累计的 300 多个,至 1998 年 11 月猛增到 16644 个,每月增长速度为 10%。