

高等学校计算机科学与技术教材

# Linux 操作系统分析教程

*Computer*

骆耀祖 主编

李 强 主审



清华大学出版社  
<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社  
<http://press.bjtu.edu.cn>



高等学校计算机科学与技术教材

# Linux 操作系统分析教程

骆耀祖 主编  
李 强 主审

清华大学出版社  
北京交通大学出版社  
·北京·

## 内 容 简 介

本书是以 Linux 作为实例的操作系统分析教程。本书对 Linux 内核 2.4 版的源代码进行了较全面的分析,既包括对中断机制、进程调度、内存管理、进程间通信、虚拟文件系统、设备驱动程序及网络子系统的分析,也包括对 Linux 整体结构的把握、Linux 的启动过程的分析及 Linux 独具特色的模块机制的分析与应用等。其中,重点剖析了 Linux 内核中最基础的部分:进程管理、内存管理及文件管理。本书可以作为高校计算机、电子工程和自动控制等相关学科的高年级本科学学生“操作系统分析”、“操作系统设计与实现”或“嵌入式系统设计”等课程的教材,也可作为系统程序员、Linux 应用开发及嵌入式系统开发人员、系统管理员及对 Linux 感兴趣的用户的参考书。本书对于那些准备进入 Linux 操作系统内部、阅读 Linux 内核源代码及在内核级进行程序开发的读者具有非常高的参考价值。

**版权所有,翻印必究。**

**本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。**

### 图书在版编目(CIP)数据

Linux 操作系统分析教程 / 骆耀祖主编. —北京:清华大学出版社;北京交通大学出版社, 2004.5

(高等学校计算机科学与技术教材)

ISBN 7-81082-241-1

I. L… II. 骆… III. Linux 操作系统 - 高等学校 - 教材 IV. TP316.81

中国版本图书馆 CIP 数据核字(2004)第 015768 号

责任编辑:谭文芳 特邀编辑:牛喘月

出版者:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686045, 62237564

印刷者:北京东光印刷厂

发行者:新华书店总店北京发行所

开本:185×260 印张:20 字数:512千字

版次:2004年5月第1版 2004年5月第1次印刷

书号:ISBN 7-81082-241-1/TP·106

印数:1~5000册 定价:27.00元

# 前 言

2000年10月,由中国计算机学会与清华大学共同主办的第六届“全国操作系统课程教学研讨会”在北京召开,来自全国各高校从事操作系统教学与研究的多名专家、教授交流了各自院校操作系统课程的教学经验与体会,介绍了国外著名大学操作系统课程的教学大纲与实施情况,对操作系统当前的发展和操作系统课程教学方式与方法的改革情况进行了充分的研讨。为了适应信息时代对高素质软件人员的需求,尽快赶上世界最新发展潮流,提高我国各大专院校的操作系统课程的整体教学水平,经过与会专家的集体讨论,会议决定把Linux作为一个重要操作系统在大学课程中开设,并决定在“百花齐放”的基础上,提高操作系统教材的编写水平。

Linux为嵌入式操作系统提供了一个极有吸引力的选择。对于计算机科学教学和科研来说, Linux具有更重要的意义。过去,一般的操作系统教材只讲操作系统的实现原理,学生既觉得抽象又感觉不到操作系统的重要价值。尽管有些书也是以UNIX为实例,但一般市场上出售的UNIX,除了价格问题之外,还不提供其核心程序的源代码,这对学生真正深入了解操作系统造成困难,更谈不上作为操作系统教学和科研的平台。而Linux提供了从内核到上层的所有软件的全部源程序代码,易于对所获得源代码的组织结构、实现原理及实现机制进行比较详细的描述,很适合做教学用操作系统。

## 本书的组织

本书是以Linux作为实例的操作系统分析教程。第1章介绍Linux的基本概念、Linux操作系统的组成及Linux内核分析方法。第2章从系统的初始化过程,具体地介绍对Linux内核源码分析的一种方法。第3章介绍Linux核心对中断的处理。第4章叙述Linux如何处理物理内存及虚拟存储技术。第5章描述进程的概念,进程之间及进程与核心之间的通信。第6章讨论Linux核心如何控制系统中的物理设备,具体是对PCI总线及设备的初始化和使用。第7章介绍虚拟文件系统的概念,讨论Linux核心的每种文件系统是如何得到支持的。第8章讨论Linux如何支持TCP/IP网络协议。第9章讨论Linux核心是如何仅在需要时动态加载某些模块。第10章讨论用户和系统内核通信的关键的枢纽——系统调用机制。第11章是本课程的课程设计指导书。附录A和附录B则提供有关Linux核心资源及Linux核心数据结构等有用信息。

考虑到Linux版本升级很快,某一版本的大量的源码分析很快就会过时。因此,本书的目标是“导读”,帮助读者对Linux内核有个概括的认识,为读者提供一个Linux如何工作的清晰的印象,而在教学过程中由学生通过自行阅读代码发现对核心算法的描述,让学生体会到认识操作系统真正在工作时的兴奋。在读完此书后再去读Linux内核源代码,会比较容易理解和掌握,这也是本书所希望达到的目标。

Linux内核在不断发展,本书描述的内容是基于2.4.xx版本的。本书写作的时候,流行的Linux发行版本所采用的Linux内核版本最高为2.4.18,而最高的稳定版本是2.6.0。

考虑到计算机专业“操作系统分析与实现”课程的实际情况,本书按教学时数45学时,实验时数16学时进行编写。

本书由骆耀祖任主编,叶宇风、何志庆任副主编。陈正铭编写了第1章,叶宇风编写了

第 3, 10 章, 骆耀祖编写了第 4, 5, 6, 8 章, 骆珍仪编写了第 7, 9 章和附录, 何志庆编写了第 2, 11 章。最后由骆耀祖和叶宇风统稿, 李强主审。

本书的编纂过程中, 参考了中国计算机科学与技术学科教程 2002 (CCC2002) 中的操作系统教学、实验内容, 并得到了清华大学出版社和北京交通大学出版社的大力支持和帮助, 在此特表示感谢! China Linux Forum 是国内最优秀的 Linux 论坛。本书的编纂过程中, 在 China Linux Forum 和其他很多站点和论坛上得到很多的知识和资源, 谨向这些站点的所有者和参与者表示真诚感谢!

由于编者才能有限, 书中可能存在不少错误, 请同行专家及读者批评指正。

编 者

2004 年 3 月

# 目 录

|                                |    |
|--------------------------------|----|
| <b>第 1 章 Linux 系统概述</b> .....  | 1  |
| 1.1 Linux 的特征及组成 .....         | 1  |
| 1.1.1 Linux 的主要特性 .....        | 1  |
| 1.1.2 Linux 的组成 .....          | 3  |
| 1.2 Linux 内核的主要组成部分 .....      | 6  |
| 1.2.1 内存管理 .....               | 7  |
| 1.2.2 进程调度 .....               | 8  |
| 1.2.3 设备驱动程序 .....             | 9  |
| 1.2.4 文件系统 .....               | 9  |
| 1.2.5 网络 .....                 | 10 |
| 1.2.6 其他 .....                 | 10 |
| 1.3 Linux 内核分析 .....           | 11 |
| 1.3.1 核心源码的组织 .....            | 12 |
| 1.3.2 Linux 内核源代码的组成 .....     | 12 |
| 1.3.3 Linux 内核分析方法 .....       | 14 |
| 练习与思考 .....                    | 16 |
| <b>第 2 章 Linux 系统的启动</b> ..... | 17 |
| 2.1 操作系统的启动 .....              | 17 |
| 2.1.1 系统引导过程简介 .....           | 17 |
| 2.1.2 硬盘结构 .....               | 18 |
| 2.1.3 引导扇区 .....               | 18 |
| 2.2 Linux 的引导过程 .....          | 20 |
| 2.2.1 Linux 系统引导过程分析 .....     | 21 |
| 2.2.2 实模式下的初始化 .....           | 22 |
| 2.2.3 保护模式下的初始化 .....          | 26 |
| 2.2.4 系统初始化 .....              | 31 |
| 练习与思考 .....                    | 33 |
| <b>第 3 章 中断和中断处理</b> .....     | 34 |
| 3.1 硬件提供的中断机制和约定 .....         | 34 |
| 3.1.1 中断产生的过程 .....            | 35 |
| 3.1.2 中断请求 .....               | 36 |
| 3.1.3 置中断标志位 .....             | 36 |
| 3.1.4 中断处理程序的不可重入性 .....       | 36 |
| 3.1.5 时钟和定时器中断 .....           | 37 |
| 3.2 Linux 的中断处理 .....          | 39 |
| 3.2.1 Linux 中断处理程序的特色 .....    | 39 |
| 3.2.2 中断的相关数据结构 .....          | 40 |
| 3.2.3 中断向量表 IDT 的初始化 .....     | 42 |

|              |                          |           |
|--------------|--------------------------|-----------|
| 3.2.4        | 中断请求队列的初始化 .....         | 46        |
| 3.3          | Linux 2.4 的软中断处理机制 ..... | 48        |
| 3.3.1        | 底半处理 .....               | 48        |
| 3.3.2        | 与底半处理相关的数据结构 .....       | 49        |
| 3.3.3        | 任务队列 .....               | 51        |
| 3.3.4        | tasklet .....            | 52        |
| 3.3.5        | 软中断 .....                | 53        |
| 3.3.6        | 使用示例 .....               | 54        |
| 3.3.7        | 讨论 .....                 | 57        |
| 3.4          | 中断处理全过程 .....            | 60        |
| 3.4.1        | 异常的处理全过程 .....           | 61        |
| 3.4.2        | 外部设备中断处理的全过程 .....       | 61        |
| 3.4.3        | 后续处理 .....               | 62        |
|              | 练习与思考 .....              | 63        |
| <b>第 4 章</b> | <b>内存管理</b> .....        | <b>64</b> |
| 4.1          | 存储层次结构和 x86 存储管理硬件 ..... | 64        |
| 4.1.1        | 内存管理基本框架 .....           | 64        |
| 4.1.2        | 地址映射的全过程 .....           | 65        |
| 4.1.3        | 地址映射的效率分析 .....          | 68        |
| 4.1.4        | 连续物理区域管理 .....           | 71        |
| 4.2          | Linux 虚存系统 .....         | 73        |
| 4.2.1        | 使用虚存的优点 .....            | 73        |
| 4.2.2        | Linux 虚存的“分页”机制 .....    | 74        |
| 4.3          | Linux 的内存页表 .....        | 75        |
| 4.3.1        | 内存页的分配和释放 .....          | 75        |
| 4.3.2        | 内存映射和需求分页 .....          | 77        |
| 4.3.3        | Linux 页缓存 .....          | 78        |
| 4.4          | 内存交换与高速缓存 .....          | 79        |
| 4.4.1        | 内存交换 .....               | 79        |
| 4.4.2        | 高速缓存 .....               | 81        |
| 4.5          | 相关系统工具 .....             | 81        |
| 4.5.1        | 建立交换空间 .....             | 82        |
| 4.5.2        | 使用交换空间 .....             | 82        |
| 4.5.3        | 分配交换空间 .....             | 83        |
| 4.5.4        | 缓冲区高速缓存的守护进程 .....       | 83        |
|              | 练习与思考 .....              | 84        |
| <b>第 5 章</b> | <b>进程管理及进程间通信</b> .....  | <b>85</b> |
| 5.1          | Linux 进程和线程 .....        | 85        |
| 5.1.1        | Linux 进程管理的数据结构 .....    | 85        |
| 5.1.2        | 标识符信息 .....              | 87        |
| 5.1.3        | 进程状态信息 .....             | 87        |
| 5.1.4        | 文件信息 .....               | 88        |
| 5.1.5        | 虚拟内存 .....               | 89        |

|              |                    |            |
|--------------|--------------------|------------|
| 5.1.6        | 时间和定时器             | 90         |
| 5.1.7        | 关于线程               | 90         |
| 5.1.8        | 会话和进程组             | 91         |
| 5.2          | 进程的创建和进程调度         | 91         |
| 5.2.1        | 进程的创建              | 91         |
| 5.2.2        | 进程的管理和调度           | 92         |
| 5.2.3        | 进程的切换              | 96         |
| 5.3          | 可执行程序              | 98         |
| 5.3.1        | 可执行可连接格式           | 98         |
| 5.3.2        | 脚本文件               | 100        |
| 5.4          | Linux 下进程间通信的主要手段  | 100        |
| 5.4.1        | 信号                 | 101        |
| 5.4.2        | 管道和套接字             | 102        |
| 5.5          | System V 的 IPC 机制  | 104        |
| 5.5.1        | 消息队列               | 104        |
| 5.5.2        | 信号量                | 105        |
| 5.5.3        | 共享内存               | 107        |
| 5.5.4        | 相关系统工具             | 108        |
| 5.6          | 内核同步机制             | 108        |
| 5.6.1        | 原子操作和信号量           | 108        |
| 5.6.2        | 任务队列               | 110        |
| 5.6.3        | 等待队列和异步信号          | 111        |
| 5.6.4        | Buzz 锁             | 115        |
| 5.6.5        | 信号灯                | 117        |
|              | 练习与思考              | 118        |
| <b>第 6 章</b> | <b>硬件和设备驱动程序</b>   | <b>120</b> |
| 6.1          | 总线                 | 120        |
| 6.1.1        | 总线概述               | 120        |
| 6.1.2        | Linux 对 PCI 总线的支持  | 121        |
| 6.2          | 设备访问方式             | 122        |
| 6.2.1        | 查询与中断              | 122        |
| 6.2.2        | 直接内存访问             | 124        |
| 6.2.3        | 内存                 | 124        |
| 6.3          | 设备驱动程序             | 125        |
| 6.3.1        | 设备驱动程序的概念          | 125        |
| 6.3.2        | 设备驱动程序的内存分配        | 126        |
| 6.3.3        | 设备驱动程序和内核的接口       | 127        |
| 6.4          | 编写 Linux 下的设备驱动程序  | 131        |
| 6.4.1        | Linux 下设备驱动程序的基本结构 | 131        |
| 6.4.2        | Linux 系统下的设备驱动程序   | 133        |
|              | 练习与思考              | 137        |
| <b>第 7 章</b> | <b>文件系统</b>        | <b>139</b> |
| 7.1          | 虚拟文件系统             | 139        |



|              |                     |            |
|--------------|---------------------|------------|
| 7.1.1        | 虚拟文件系统概述            | 139        |
| 7.1.2        | 基本对象与方法             | 140        |
| 7.1.3        | 文件系统的注册、安装与卸载       | 141        |
| 7.1.4        | 虚拟文件系统的实现函数和数据结构    | 144        |
| 7.2          | 高速缓存                | 146        |
| 7.2.1        | 文件系统分配机制            | 147        |
| 7.2.2        | 块高速缓存               | 148        |
| 7.2.3        | 索引节点高速缓存            | 149        |
| 7.2.4        | 目录高速缓存              | 149        |
| 7.3          | /proc 文件系统          | 150        |
| 7.3.1        | 概述                  | 150        |
| 7.3.2        | /proc 文件系统的注册       | 150        |
| 7.3.3        | 建立/proc 文件系统的超级块    | 151        |
| 7.3.4        | /proc 文件系统超级块的操作函数集 | 153        |
| 7.3.5        | /proc 文件的管理         | 155        |
| 7.3.6        | /proc 文件的默认操作       | 160        |
| 7.4          | EXT2 文件系统           | 169        |
| 7.4.1        | EXT2 索引节点           | 170        |
| 7.4.2        | EXT2 超块             | 171        |
| 7.4.3        | EXT2 块组描述符          | 172        |
| 7.4.4        | EXT2 目录             | 172        |
| 7.4.5        | EXT2 文件系统中数据块的分配和释放 | 173        |
| 7.4.6        | 日志文件系统              | 174        |
|              | 练习与思考               | 175        |
| <b>第 8 章</b> | <b>网络</b>           | <b>177</b> |
| 8.1          | 网络协议                | 177        |
| 8.1.1        | TCP/IP 协议概述         | 177        |
| 8.1.2        | TCP/IP 协议工作原理及数据流   | 178        |
| 8.2          | 套接字                 | 180        |
| 8.2.1        | 套接字的类型              | 180        |
| 8.2.2        | 套接字的基本结构和工作原理       | 181        |
| 8.2.3        | 套接字的初始化             | 183        |
| 8.2.4        | 套接字的通信过程            | 193        |
| 8.3          | 套接字缓冲区              | 198        |
| 8.3.1        | 套接字缓冲区操作基本原理        | 199        |
| 8.3.2        | 套接字缓冲区的数据结构         | 199        |
| 8.3.3        | 套接字缓冲区提供的函数         | 200        |
| 8.3.4        | 套接字缓冲区的上层支持例程       | 202        |
| 8.4          | 网络设备接口源码            | 203        |
| 8.4.1        | Linux 网络接口源码结构      | 203        |
| 8.4.2        | 网络设备接口              | 205        |
| 8.4.3        | 网络接口核心              | 207        |
| 8.4.4        | 网络协议接口              | 208        |

|               |                           |            |
|---------------|---------------------------|------------|
| 8.4.5         | Socket 层和用户的衔接 .....      | 212        |
| 8.5           | IP 层 .....                | 212        |
| 8.5.1         | 接收 IP 数据包 .....           | 212        |
| 8.5.2         | 发送 IP 数据包 .....           | 213        |
| 8.5.3         | 数据包的分段和重组 .....           | 213        |
| 8.5.4         | 地址解析协议 .....              | 214        |
| 8.5.5         | IP 路由 .....               | 215        |
|               | 练习与思考 .....               | 217        |
| <b>第 9 章</b>  | <b>模块的动态加载和系统配置</b> ..... | <b>219</b> |
| 9.1           | 模块的动态加载 .....             | 219        |
| 9.1.1         | 模块的加载 .....               | 220        |
| 9.1.2         | 模块的卸载 .....               | 221        |
| 9.1.3         | 内核模块的管理 .....             | 222        |
| 9.2           | Linux 内核配置系统 .....        | 223        |
| 9.2.1         | 配置系统的基本结构 .....           | 223        |
| 9.2.2         | Makefile .....            | 223        |
| 9.2.3         | 配置文件 .....                | 228        |
| 9.2.4         | 配置实例 .....                | 231        |
|               | 练习与思考 .....               | 233        |
| <b>第 10 章</b> | <b>系统调用</b> .....         | <b>235</b> |
| 10.1          | 系统调用概述 .....              | 235        |
| 10.1.1        | 系统调用过程 .....              | 235        |
| 10.1.2        | 系统调用的进入 .....             | 236        |
| 10.1.3        | 系统调用总控程序 .....            | 237        |
| 10.1.4        | 一个简单的系统调用例子 .....         | 238        |
| 10.2          | 与进程管理相关的系统调用 .....        | 242        |
| 10.2.1        | getpid .....              | 242        |
| 10.2.2        | fork .....                | 243        |
| 10.2.3        | exit .....                | 244        |
| 10.2.4        | exit 和 _exit .....        | 245        |
| 10.2.5        | 其他一些常见的系统调用 .....         | 247        |
| 10.3          | 僵尸进程 .....                | 247        |
| 10.3.1        | wait .....                | 248        |
| 10.3.2        | waitpid .....             | 250        |
| 10.3.3        | exec 系统调用 .....           | 252        |
| 10.3.4        | 稍稍深入 .....                | 252        |
| 10.4          | 与进程管理相关的系统调用实战 .....      | 256        |
| 10.4.1        | Mini Shell .....          | 256        |
| 10.4.2        | Daemon 进程 .....           | 259        |
| 10.5          | 为内核增加系统调用实战 .....         | 261        |
| 10.5.1        | 增加系统调用的两种方法 .....         | 261        |
| 10.5.2        | 相关内核源代码分析 .....           | 262        |
| 10.5.3        | 对内核源码的修改 .....            | 263        |

|                                     |            |
|-------------------------------------|------------|
| 练习与思考                               | 265        |
| <b>第 11 章 课程设计指导书</b>               | <b>266</b> |
| 11.1 Linux 启动代码的分析                  | 267        |
| 11.2 Linux 的 fork, exec, wait 函数的分析 | 269        |
| 11.3 Linux 的消息函数的分析                 | 273        |
| 11.4 Linux 的调度和时钟中断处理代码的分析          | 276        |
| 11.5 自选课题                           | 279        |
| 11.5.1 Linux 的文件系统                  | 279        |
| 11.5.2 Linux 的存储管理                  | 279        |
| 11.5.3 Linux 的设备管理                  | 279        |
| 11.5.4 Linux 进程间通信                  | 280        |
| <b>附录 A Linux 核心数据结构</b>            | <b>281</b> |
| <b>附录 B Linux 系统调用列表</b>            | <b>302</b> |
| <b>参考文献</b>                         | <b>309</b> |

# 第 1 章 Linux 系统概述

Linux 作为一个多任务、多用户的操作系统，以其很好的稳定性赢得了广大用户，并迅速发展成为操作系统中的主流。本章简单介绍 Linux 的基本概念，Linux 操作系统的组成及 Linux 内核分析方法。

## 1.1 Linux 的特征及组成

Linux 系统是包含内核、系统工具、完整的开发环境和应用的类 UNIX 操作系统。这个系统是由世界各地的成千上万的程序员设计和实现的。1984 年，Richard Stallman 创立了 GNU 工程，其目标是开发一个完全免费的类 UNIX 系统及其应用程序。1991 年，芬兰赫尔辛基大学一位名叫 Linus Torvalds 的学生开始了开放源代码的 Linux 雏形的设计。其目的是建立不受任何商品化软件的版权制约的、全世界都能自由使用的 UNIX 兼容产品。由于 Linux 内核及多数应用程序都基于 GPL (GNU 公共许可协议)，且具有结构清晰、功能简捷等特点，逐渐成为一个稳定可靠、功能完善的操作系统。

由于 Linux 是一套具有 UNIX 全部功能的免费操作系统，它在众多的软件中占有很大的优势，为广大的计算机爱好者提供了学习、探索及修改计算机操作系统内核的机会。不管运行在何种平台上，95% 的 Linux 核心代码都是相同的。因此，本书关于 Linux 内核的讨论内容 95% 以上是和机器无关的。

### 1.1.1 Linux 的主要特性

#### 1. 开放性

开放性是指系统遵循开放系统互连 (Open Systems Interconnection, OSI) 国际标准。凡遵循国际标准所开发的硬件和软件，都能彼此兼容，可以方便地实现互连。

#### 2. 多用户

多用户是指系统资源可被不同用户各自拥有使用，即每个用户对自己的资源 (例如：文件、设备) 有特定的权限，互不影响。Linux 和 UNIX 都具有多用户的特性。

#### 3. 多任务

多任务是现代计算机的最主要的一个特点。它是指计算机同时执行多个程序，而且各个程序的运行互相独立。Linux 系统调度每个进程平等地访问微处理器。由于 CPU 的处理速度非常快，其结果是，启动的应用程序看起来好像在并行运行。事实上，从处理器执行一个应用程序中的一组指令到 Linux 调度微处理器再次运行这个程序之间只有很短的时间延迟，

用户是感觉不出来的。

#### 4. 良好的用户界面

Linux 向用户提供了两种界面：用户界面和系统调用。Linux 的传统用户界面基于文本的命令行界面，即 shell，它既可以联机使用，又可存放在文件上脱机使用。shell 有很强的程序设计能力，用户可方便地用它编制程序，从而为用户扩充系统功能提供更高级的手段。可编程 shell 是指将多条命令组合在一起，形成一个 shell 程序；这个程序可以单独运行，也可以与其他程序同时运行。

系统调用为用户提供编程时使用的界面。用户可在编程时直接使用系统提供的系统调用命令。系统通过这个界面为用户程序提供低级、高效率的服务。

Linux 还为用户提供图形用户界面。它利用鼠标、菜单、窗口、滚动条等设施，给用户呈现一个直观、易操作、交互性强的友好的图形化界面。

#### 5. 设备独立性

设备独立性是指操作系统把所有外部设备统一当成文件来看待，只要安装它们的驱动程序，任何用户都可以像使用文件一样，操纵、使用这些设备，而不必知道它们的具体存在形式。

具有设备独立性的操作系统，通过把每个外围设备看做一个独立文件来简化增加新设备的工作。当需要增加新设备时，系统管理员就在内核中增加必要的连接。这种连接（也称设备驱动程序）保证每次调用设备提供服务时，内核以相同的方式对它们进行处理。当新的更好的外设被开发并交付用户时，操作允许在这些设备连接到内核后，就能不受限制地立即进行访问。设备独立性的关键在于内核的适应能力。其他操作系统只允许一定数量或一定种类的外部设备连接。设备独立性的操作系统能够容纳任意种类及任意数量的设备，因为每个设备都通过其与内核的专用连接独立进行访问。

Linux 是具有设备独立性的操作系统，它的内核具有高度适应能力，随着更多的程序员加入 Linux 编程，会有更多硬件设备加入各种 Linux 内核和发行版本中。由于用户可以免费得到 Linux 的内核源代码，因此用户可以修改内核源代码，以便适应新增加的外部设备。

#### 6. 丰富的网络功能

完善的内置网络是 Linux 的一大特点。Linux 在通信和网络功能方面优于其他操作系统。其他操作系统不包含如此紧密地和内核结合在一起的连接网络的能力，也没有内置这些连网特性的灵活性。而 Linux 为用户提供了完善的、强大的网络功能。

支持 Internet 是其网络功能之一。Linux 免费提供大量支持 Internet 的软件，Internet 是在 UNIX 领域中建立并繁荣起来的，在这方面使用 Linux 是相当方便的，用户能用 Linux 与世界上的其他人通过 Internet 网络进行通信。

文件传输是其网络功能之二。用户能够通过一些 Linux 命令完成内部信息或文件的传输。

远程访问是其网络功能之三。Linux 不仅允许进行文件和程序的传输，还为系统管理员和技术人员提供访问其他系统的窗口。通过这种远程访问的功能，一位技术人员能够有效地为多个远程系统服务。

## 7. 可靠的系统安全

Linux 采取了许多安全技术措施,包括对读、写进行权限控制、带保护的子系统、审计跟踪、核心授权等。这为网络多用户环境中的用户提供了必要的安全保障。

## 8. 良好的可移植性

可移植性是指将操作系统从一个平台转移到另一个平台,仍然能按其自身的方式运行的能力。

Linux 是一种可移植的操作系统,能够在从微型计算机到大型计算机的任何环境中在任何平台上运行。可移植性为运行 Linux 的不同计算机平台与其他任何机器进行准确而有效的通信提供了手段,不需要另外增加特殊的和昂贵的通信接口。

### 1.1.2 Linux 的组成

通常操作系统由内核和一些系统服务程序(命令解释、库文件、链接和编译程序等)组成。内核是操作系统的灵魂,它为用户进程提供一个虚拟机接口。用户进程可以并行运行、公平地占用系统资源而互不干扰。

Linux 一般由内核、shell、文件结构和实用工具四个主要部分组成。

#### 1. Linux 内核

内核是系统的核心,是运行程序和管理像磁盘和打印机等硬件设备的核心程序。它从用户那里接受命令并把命令送给内核去执行。

#### 2. Linux shell

shell 是系统的用户界面,提供用户与内核进行交互操作的一种接口。它接收用户输入的命令并且把它送入内核去执行。

shell 实际上是一个命令解释器,它解释由用户输入的命令并且把它们送到内核。不仅如此,shell 有自己的编程语言用于对命令的编辑,它允许用户编写由 shell 命令组成的程序。shell 编程语言具有普通编程语言的很多特点,如它也有循环结构和分支控制结构等,用这种编程语言编写的 shell 程序与其他应用程序具有同样的效果。

Linux 提供像 Microsoft Windows 那样的可视的命令输入界面——X Windows 的图形用户界面(GUI)。它提供很多窗口管理器,其操作就像 Windows 一样,有窗口、图标和菜单,所有的管理都通过鼠标控制。现在比较流行的窗口管理器是 KDE 和 GNOME。

每个 Linux 系统的用户可以拥有他自己的用户界面或 shell,用以满足他自己专门的 shell 需要。

#### 3. Linux 文件结构

文件结构是文件存放在磁盘等存储设备上的组织,主要体现在对文件和目录的组织上。目录提供管理文件的一个方便而有效的途径,能从一个目录切换到另一个目录,而且可以设置目录和文件的权限,设置文件的共享程度。

大多数 Linux 发行版本文件系统的目录树布局采用多级树形结构，遵循文件系统标准 FSSTND (File System Standard)。遵循这一标准有利于编写或移植软件，同时也有利于进行系统管理和维护。

完整的目录树可以划分为小的部分，这些小部分又可以单独存放在自己的磁盘或分区上。这样，相对稳定的部分和经常变化的部分可以单独存放在不同的分区中，从而方便备份或系统管理。目录树的主要部分有 root, /usr, /var, /home 等 (图 1-1)。这样的布局可以方便在 Linux 计算机之间共享文件系统的某些部分。用户可以浏览整个系统，可以进入任何一个已授权进入的目录，访问那里的文件。使用 Linux，用户可以设置目录和文件的权限，以便允许或拒绝其他人对其进行访问。

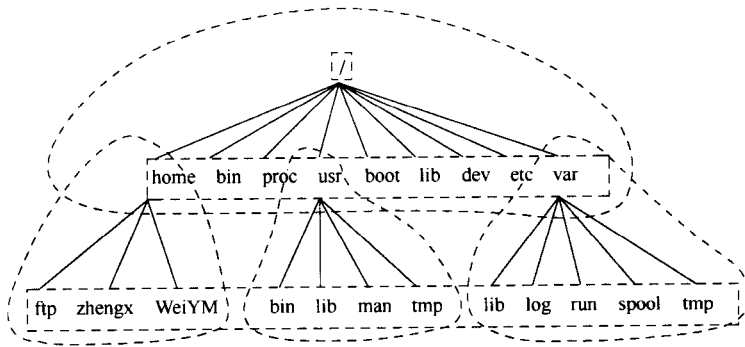


图 1-1 Linux 目录树的不同部分

表 1-1 给出了目录树的不同部分在 Linux 系统中所扮演的角色。

表 1-1 目录树的不同部分在 Linux 系统中所扮演的角色

| 名称    | 角色   |
|-------|--|
| root  | root 文件系统中包含和特定计算机相关的内容，因此，root 文件系统一般保存在计算机的本地硬盘中，但也可以保存在 ramdisk 或网络驱动器中。root 文件系统中的内容包括引导系统的必备文件，文件系统的挂载信息及系统修复工具和备份工具等                 |
| /usr  | /usr 文件系统中包含通常操作中不需要进行修改的命令程序文件、程序库、手册和其他文档等。由于 /usr 中的文件并不和特定的计算机相关，也不会通常在的使用中修改，因此可通过网络共享这些文件。这样，当 root 安装了新的软件之后，所有共享这个文件系统的计算机均可使用新的软件 |
| /var  | /var 文件系统中包含经常变化的文件，例如打印机、邮件、新闻等的假脱机目录、日志文件、格式化后的手册页及临时文件等。按照传统习惯，/var 的文件通常包含在 /usr 中，但这样就无法将 /usr 文件系统挂载为只读文件系统                          |
| /home | /home 中包含用户的主目录，用户的数据保存在其主目录中，如果有必要，也可将 /home 划分为不同的文件系统，例如 /home/students 和 /home/teachers 等。将 /home 放置在单独的文件系统中便于进行用户数据的备份               |
| /proc | /proc 文件系统并不保存在磁盘上，相反，操作系统在内存中创建这个文件系统。其中包含一些和系统相关的信息，例如 CPU、DMA 通道及中断的使用信息等   |

表 1-1 中，目录树的不同部分又称文件系统，它们不必保存在单独的文件系统中。

(1) root 文件系统

root 文件系统中包含一些关键文件，同时其内容也比较小。如果 root 文件系统被破坏，操作系统就无法正确引导。root 文件系统中包含的文件和目录见表 1-2。

表 1-2 root 文件系统中的文件和目录

| 名 称                         | 说 明  |
|-----------------------------|--|
| /vmlinuz                    | 系统的标准引导映像,通常以压缩形式出现  |
| /bin                        | 包含引导过程必需的命令,也可由普通用户使用  |
| /sbin                       | 和/bin类似,尽管其中的命令可由普通用户使用,但由于这些命令属于系统级命令,因此无特殊需求不使用其中的命令             |
| /etc                        | 包含与特定计算机相关的配置文件  |
| /root                       | root 用户的主目录  |
| /lib                        | root 文件系统中的程序要使用的共享库保存在该目录中  |
| /lib/modules                | 包含可装载的内核模块   |
| /dev                        | 包含设备文件   |
| /tmp                        | 包含临时文件。引导后运行的程序应当在/var/tmp中保存文件,因为其中的可用空间大一些                       |
| /boot                       | 包含引导装载程序要使用的文件。内核映像通常保存在这个目录中。因为多个内核映像会占用很多磁盘空间,因此可将该目录放置在单独的文件系统中 |
| /mnt                        | 临时文件系统的挂装点   |
| /usr, /var,<br>/home, /proc | 其他文件系统的挂装点   |

### (2) /usr 文件系统

/usr 文件系统中包含所有的程序文件及联机文档,因此其内容通常很大。/usr 文件系统中包含的文件和目录见表 1-3。

表 1-3 /usr 文件系统中的目录

| 目 录                           | 说 明                                  |
|-------------------------------|--------------------------------------|
| /usr/X11R6                    | 包含 X 窗口系统的所有文件                       |
| /usr/X386                     | 和/usr/X11R6类似,但包含 X11 的 Release 5    |
| /usr/bin                      | 绝大多数用户命令。其他命令包含在/bin和/usr/local/bin中 |
| /usr/sbin                     | root 文件中不需要的系统管理命令                   |
| /usr/man, /usr/info, /usr/doc | 分别包含手册页、GNU Info 文档及其他杂项文档           |
| /usr/include                  | C 语言的头文件                             |
| /usr/lib                      | 程序和子系统所使用的不变的数据文件                    |
| /usr/local                    | 本地挂装的软件和其他文件的存放位置                    |

### (3) /var 文件系统

/var 包含系统运行过程中经常发生变化的文件。/var 文件系统中包含的目录见表 1-4。

表 1-4 /var 文件系统中的目录

| 目 录         | 说 明                      |
|-------------|--------------------------|
| /var/catman | 格式化手册页的高速缓存              |
| /var/lib    | 包含系统运行时经常改变的文件           |
| /var/local  | 安装/usr/local中的程序的可变数据    |
| /var/lock   | 包含锁文件                    |
| /var/log    | 包含程序产生的日志文件              |
| /var/run    | 该目录包含在下次引导之前有效的,和系统相关的信息 |
| /var/spool  | 排队任务的假脱机目录               |
| /var/tmp    | 包含大的临时文件,或者保存时间较长的临时文件   |



#### (4) /proc 文件系统

/proc 文件系统并不保存在磁盘上，操作系统在内存中创建这个文件系统。/proc 文件系统中包含的文件和目录见表 1-5。

表 1-5 /proc 文件系统中的文件和目录

| 名 称               | 说 明  |
|-------------------|--|
| /proc/1           | 该目录中包含进程号为 1 的进程信息。每个进程在 /proc 目录下有一个以自己的进程号为名称的目录 |
| /proc/cpuinfo     | 有关 CPU 名称、型号、性能和类型的信息                              |
| /proc/devices     | 当前内核中的设备驱动程序列表                                     |
| /proc/dma         | 当前使用的 DMA 通道                                       |
| /proc/filesystems | 内核支持的文件系统  |
| /proc/interrupts  | 当前使用的中断信息  |
| /proc/ioports     | 当前使用的 I/O 端口                                       |
| /proc/kcore       | 系统物理内存的映像  |

文件结构的相互关联性使共享数据变得容易，几个用户可以访问同一个文件。Linux 是一个多用户系统，操作系统本身的驻留程序存放在以根目录开始的专用目录中，有时被指定为系统目录。图 1-1 中那些根目录下的目录就是系统目录。

内核、shell 和文件结构一起形成基本的操作系统结构。它们使得用户可以运行程序，管理文件及使用系统。此外，Linux 操作系统还有许多被称为实用工具的程序，辅助用户完成一些特定的任务。

### 4. Linux 实用工具

标准的 Linux 系统都有一套叫做实用工具的程序。它们是专门的程序，例如编辑器、执行标准的计算操作等。用户也可以产生自己的工具。

## 1.2 Linux 内核的主要组成部分

操作系统一般由内核和一些系统程序组成，还有一些应用程序帮助用户完成特定任务。内核是操作系统的灵魂，它负责管理磁盘上的文件、内存，负责启动并运行程序，负责从网络上接收和发送数据包，等等。总而言之，操作系统实际是抽象的资源操作到具体硬件操作细节之间的接口。对 Linux 这样的多用户操作系统来说，它还需要避免用户对硬件的直接访问，防止用户之间的互相干扰。

系统程序及其他所有的程序在内核之上运行，程序和内核之间的接口由操作系统提供的一组“抽象指令”定义，这些抽象指令称为“系统调用”。所有运行在内核之上的程序可以分为系统程序和用户程序两大类，但它们统统运行在“用户模式”之下。系统程序和用户程序之间的界限是模糊的。系统程序一般指运行系统所不可缺少的程序，例如 Linux 中的 shell；而用户程序则是给用户特定功能的程序，例如字处理程序或游戏程序。实际的操作系统中往往还包含一些工具程序（如编译器）及一些联机文档。

Linux 系统的内核采用单块结构，可以动态地加载和卸载模块。系统利用内核模块的可