

# Linux

## 内核分析与实例应用

博嘉科技 主编 代玲莉 欧阳劲 编著

 时代自由软件系列

# Linux 内核分析与 实例应用

博 嘉 科 技 主 编  
代 玲 莉 欧 阳 劲 编 著

国防工业出版社

·北京·

## 内 容 简 介

本书通过分析 Linux 内核源代码,对 Linux 的内核系统组成做了详尽的讲解。主要包括 Linux 的进程管理及进程间通信机制、内存管理、文件系统、网络、设备管理和内核机制等内容,同时辅以实例来帮助读者熟悉和掌握 Linux 内核原理。

本书结构清晰、实例丰富、语言流畅,它既可供 Linux 初学者和对 Linux 有一定了解的读者学习使用,也可以作为资深 Linux 研究开发人员的参考指南。

### 图书在版编目(CIP)数据

Linux内核分析与实例应用/代玲莉,欧阳劲编著.

北京:国防工业出版社,2002.9

(e时代自由软件系列)

ISBN 7-118-02836-3

I.L... II.①代... ②欧... III.Linux 操作系统—  
程序分析 IV.TP316.89

中国版本图书馆 CIP 数据核字(2002)第 018779 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

\*

开本 787×1092 1/16 印张 20 455 千字

2002 年 9 月第 1 版 2002 年 9 月北京第 1 次印刷

印数:1—4000 册 定价:27.00 元

(本书如有印装错误,我社负责调换)

# 前 言

Linux 随着互联网的普及而迅速遍及全球,成为一种很受欢迎的操作系统。它是一个性能稳定、功能强大、效率高的操作系统,能实现多任务、多用户、多平台的操作,而且对硬件配置的要求相对较低。

自由和开放是 Linux 最本质的特征。因为自由,我们可以通过互联网免费下载 Linux 的最新核心版本和各种 Linux 的软件工具,这与“贵族化”的 UNIX 相比尤显突出;因为开放,我们可以深入 Linux 系统的内部世界,通过剖析其内核源代码和模块组织,进行二次开发应用,而且任何人都可以对内核源代码进行修改和增补,发表在互联网上。可以说, Linux 是互联网的产物,其自由和开放的特征也因为互联网得到了更好的体现。

## ◆ 本书主要内容

内核是 Linux 操作系统的灵魂。本文旨在通过分析 Linux 源代码,使读者熟悉和掌握 Linux 内核,深入了解底层的内容。为了进一步帮助读者学习 Linux,本文还列举了大量的应用实例,这就避免了在学习过程中对枯燥无味的内核理论产生的厌烦情绪。

本书的主要内容包括:

第 1 章主要介绍了 Linux 的起源、发展、特性及其系统组成和源代码的相关知识。

第 2 章主要分析了 Linux 的进程管理。包括进程的控制、调度和有关进程的其他内容。

第 3 章主要讲解了 Linux 进程间通信机制,分析 Linux 信号、管道、System V 进程间通信机制、消息队列、信号量、共享内存等内容。

第 4 章主要分析了 Linux 的内存管理功能。内存管理是 Linux 内核最复杂的任务之一,它涉及到 Linux 的内存映射、缺页处理、交换机制、内存的分配和回收、缓存和刷新以及共享内存等。

第 5 章主要是 Linux 的文件系统,包括 EXT2 文件系统和 VFS 虚拟文件系统。同时,还介绍了对文件的基本操作。

第 6 章主要分析了 Linux 的网络管理。讲解 BSD 套接字、INET 套接字层以及 IP 层和网络的其他知识。

第 7 章主要分析了 Linux 的设备管理。通过介绍 PCI 总线、设备中的中断、内存访问、硬盘和设备驱动程序的知识,使读者能编写简单的驱动程序。

第 8 章主要介绍了 Linux 的内核机制和内核模块。

第 9 章主要分析了 Linux 的开机初始化过程。

在附录中列出了在本书中提到的 Linux 内核源代码中的数据结构。

此书所用的源代码是 Linux 内核 2.4.7 版,书中所举的例子也是在此内核中进行调

试、编译的。

## ◆ 适用对象

本书要求读者具有一定的 C 语言基础。可供 Linux 初学者和对 Linux 有一定了解的读者学习使用,也可以作为有一定经验的 Linux 研究开发人员的参考读物。

## ◆ 本书编写分工

本书的出版,曾得到了编者的父母和许多朋友的鼓励和支持,特别是王松老师,他为编者提供了出版此书的机会。

本书由代玲莉、欧阳劲编著。参与编写本书的还有:代林晴、李明、代云峰、赵华勇、赵丽萍、代凡、岳海波、杨宏颖、祝懿、冯莉芳、贺利芳、赵春芬、申辉贤、朱西平、蔡海慧、张连波、陆阿妮、袁妮、毕长玉、文涛、崔敏、王立飞、黄俊娇、赵巧玲、邹云川、黄祥志、杜建华、雷继明、邓本涛、谢先波、黄成刚。最后,还要感谢本书的美工和整理人员,他们也为本书付出了辛勤的劳动。

对于本书,虽然我们都全力以赴进行编写,但个人的学识、能力毕竟有限,如果发现了书中的谬误,欢迎通过各种方式批评指正并提出宝贵意见和建议。若有意见或者建议,欢迎与我们联系,Email:bojiakeji@163.net。我们的通讯地址是:四川大学西区建筑学院成都博嘉科技资讯有限公司,邮编 610065。

若读者、网友发现有网站未经授权而转载本书内容或提供各种形式的下载服务,请予举报,经查属实,将予以重奖。联系电话:(028)5404228、5460593。

编 者

# 目 录

|                              |    |
|------------------------------|----|
| <b>第 1 章 概论</b> .....        | 1  |
| 1.1 初识 Linux .....           | 1  |
| 1.1.1 Linux 的诞生 .....        | 1  |
| 1.1.2 Linux 的发展 .....        | 1  |
| 1.2 Linux 操作系统结构 .....       | 2  |
| 1.2.1 Linux 的特性 .....        | 2  |
| 1.2.2 Linux 的系统组成 .....      | 3  |
| 1.2.3 Linux 与其他操作系统的区别 ..... | 5  |
| 1.3 Linux 的内核源代码 .....       | 5  |
| 1.3.1 如何得到源代码 .....          | 5  |
| 1.3.2 源代码的目录体系 .....         | 5  |
| 1.3.3 源代码的解读 .....           | 6  |
| 1.4 本章小结 .....               | 8  |
| <b>第 2 章 进程管理</b> .....      | 9  |
| 2.1 Linux 进程 .....           | 9  |
| 2.1.1 数据结构 task_struct ..... | 9  |
| 2.1.2 标识符 .....              | 12 |
| 2.1.3 时间和定时器 .....           | 13 |
| 2.2 Linux 进程控制 .....         | 14 |
| 2.2.1 创建进程 .....             | 14 |
| 2.2.2 执行进程 .....             | 18 |
| 2.2.3 等待进程 .....             | 21 |
| 2.2.4 结束进程 .....             | 23 |
| 2.3 程序执行 .....               | 26 |
| 2.3.1 写时复制 .....             | 26 |
| 2.3.2 ELF 可执行文件 .....        | 26 |
| 2.4 Linux 进程调度 .....         | 28 |
| 2.4.1 调度原理 .....             | 28 |
| 2.4.2 多处理器系统中的调度 .....       | 30 |
| 2.4.3 Linux 进程调度的实现 .....    | 31 |

|                           |           |
|---------------------------|-----------|
| 2.4.4 重要函数分析·····         | 35        |
| 2.5 Linux 进程管理实例·····     | 42        |
| 2.6 本章小结·····             | 48        |
| <b>第3章 进程间通信机制</b> ·····  | <b>49</b> |
| 3.1 信号·····               | 49        |
| 3.1.1 信号机制·····           | 49        |
| 3.1.2 信号处理·····           | 51        |
| 3.1.3 实例·····             | 54        |
| 3.2 管道·····               | 55        |
| 3.2.1 管道的实现机制·····        | 56        |
| 3.2.2 创建管道·····           | 57        |
| 3.2.3 命名管道·····           | 59        |
| 3.3 System V 进程间通信机制····· | 64        |
| 3.3.1 消息队列·····           | 65        |
| 3.3.2 信号量·····            | 71        |
| 3.3.3 共享内存·····           | 77        |
| 3.4 本章小结·····             | 81        |
| <b>第4章 内存管理</b> ·····     | <b>82</b> |
| 4.1 Linux 内存管理体系结构·····   | 82        |
| 4.1.1 Linux 的地址空间·····    | 82        |
| 4.1.2 地址转换技术·····         | 82        |
| 4.1.3 Linux 内存管理机制·····   | 83        |
| 4.2 Linux 分页机制·····       | 84        |
| 4.2.1 页目录和页表·····         | 84        |
| 4.2.2 页的分配和回收·····        | 85        |
| 4.3 内存映射·····             | 88        |
| 4.4 缺页处理·····             | 90        |
| 4.5 交换机制·····             | 92        |
| 4.5.1 物理页的换出·····         | 92        |
| 4.5.2 页的换出和淘汰策略·····      | 96        |
| 4.6 内存分配和回收·····          | 96        |
| 4.6.1 分配与回收策略·····        | 97        |
| 4.6.2 相关函数与具体操作·····      | 97        |
| 4.6.3 应用实例·····           | 98        |
| 4.7 Linux 缓存与刷新机制·····    | 99        |
| 4.7.1 Linux 系统中的缓存·····   | 99        |
| 4.7.2 刷新机制·····           | 101       |

|                           |            |
|---------------------------|------------|
| 4.8 共享内存 .....            | 102        |
| 4.9 本章小结 .....            | 102        |
| <b>第5章 文件系统 .....</b>     | <b>103</b> |
| 5.1 文件系统概述 .....          | 103        |
| 5.1.1 文件系统的发展 .....       | 103        |
| 5.1.2 文件系统描述 .....        | 104        |
| 5.1.3 文件描述符 .....         | 106        |
| 5.2 EXT2 文件系统 .....       | 108        |
| 5.2.1 EXT2 文件的特性 .....    | 108        |
| 5.2.2 EXT2 文件的索引节点 .....  | 110        |
| 5.2.3 EXT2 文件的超级块 .....   | 112        |
| 5.2.4 EXT2 文件的组描述符 .....  | 115        |
| 5.2.5 EXT2 文件的目录 .....    | 116        |
| 5.2.6 文件的查询 .....         | 117        |
| 5.3 虚拟文件系统 .....          | 118        |
| 5.3.1 VFS 的超级块 .....      | 118        |
| 5.3.2 VFS 索引节点 .....      | 120        |
| 5.3.3 Linux 文件的控制 .....   | 123        |
| 5.4 文件系统缓存 .....          | 127        |
| 5.4.1 VFS 中的索引节点缓存 .....  | 128        |
| 5.4.2 VFS 中的目录缓存 .....    | 128        |
| 5.4.3 缓冲区缓存 .....         | 128        |
| 5.5 /proc 文件系统 .....      | 131        |
| 5.6 特殊设备文件 .....          | 132        |
| 5.7 Linux 文件系统操作及应用 ..... | 133        |
| 5.7.1 文件的打开与关闭 .....      | 133        |
| 5.7.2 文件的读写 .....         | 135        |
| 5.7.3 文件的定位 .....         | 136        |
| 5.7.4 文件的查询 .....         | 137        |
| 5.7.5 改变文件的属性 .....       | 137        |
| 5.7.6 目录文件操作 .....        | 138        |
| 5.8 Linux 文件管理实例 .....    | 139        |
| 5.9 本章小结 .....            | 148        |
| <b>第6章 网络管理 .....</b>     | <b>149</b> |
| 6.1 TCP/IP 网络 .....       | 149        |
| 6.1.1 网络结构 .....          | 149        |
| 6.1.2 Linux 中的网络管理 .....  | 150        |

|              |                    |            |
|--------------|--------------------|------------|
| 6.2          | BSD 套接字接口 .....    | 150        |
| 6.2.1        | BSD 套接字 .....      | 151        |
| 6.2.2        | socket 通信 .....    | 152        |
| 6.3          | INET 套接字层 .....    | 154        |
| 6.3.1        | 创建 BSD 套接字 .....   | 154        |
| 6.3.2        | 绑定地址 .....         | 156        |
| 6.3.3        | 连接 .....           | 157        |
| 6.3.4        | 监听 .....           | 158        |
| 6.3.5        | 接受连接请求 .....       | 159        |
| 6.4          | IP 层 .....         | 160        |
| 6.4.1        | 套接字缓冲区 .....       | 160        |
| 6.4.2        | 数据包的发送 .....       | 165        |
| 6.4.3        | 数据包的接收 .....       | 166        |
| 6.4.4        | 数据包的分段 .....       | 167        |
| 6.5          | 地址解析协议 .....       | 168        |
| 6.6          | IP 路由 .....        | 169        |
| 6.6.1        | 路由缓存 .....         | 169        |
| 6.6.2        | 转发信息数据库 .....      | 170        |
| 6.7          | Linux 网络管理实例 ..... | 170        |
| 6.8          | 本章小结 .....         | 184        |
| <b>第 7 章</b> | <b>设备管理 .....</b>  | <b>185</b> |
| 7.1          | Linux 设备管理概述 ..... | 185        |
| 7.2          | PCI 总线 .....       | 187        |
| 7.2.1        | PCI 总线结构 .....     | 187        |
| 7.2.2        | PCI 总线的地址管理 .....  | 189        |
| 7.2.3        | PCI 桥 .....        | 190        |
| 7.2.4        | PCI 数据结构 .....     | 190        |
| 7.3          | 设备中的内存管理 .....     | 192        |
| 7.3.1        | 设备内存地址 .....       | 192        |
| 7.3.2        | 对设备内存的操作 .....     | 193        |
| 7.3.3        | 直接内存访问 .....       | 194        |
| 7.4          | 中断 .....           | 195        |
| 7.4.1        | 可编程中断控制器 .....     | 196        |
| 7.4.2        | Linux 中断门 .....    | 197        |
| 7.4.3        | 硬件中断处理 .....       | 198        |
| 7.4.4        | 软中断 .....          | 201        |
| 7.5          | Linux 设备驱动程序 ..... | 203        |
| 7.5.1        | 设备驱动程序 .....       | 203        |

|              |                          |            |
|--------------|--------------------------|------------|
| 7.5.2        | 字符设备驱动程序 .....           | 205        |
| 7.5.3        | 块设备驱动程序 .....            | 208        |
| 7.6          | 硬盘 .....                 | 212        |
| 7.6.1        | 硬盘分区 .....               | 212        |
| 7.6.2        | 硬盘驱动程序 .....             | 213        |
| 7.7          | 网络设备 .....               | 214        |
| 7.8          | 设备驱动程序的编程实例 .....        | 215        |
| 7.9          | 本章小结 .....               | 224        |
| <b>第 8 章</b> | <b>内核机制与内核模块 .....</b>   | <b>225</b> |
| 8.1          | 内核机制 .....               | 225        |
| 8.1.1        | Bottom Half 控制 .....     | 225        |
| 8.1.2        | 任务队列 .....               | 228        |
| 8.1.3        | 定时器 .....                | 228        |
| 8.1.4        | 等待队列 .....               | 229        |
| 8.1.5        | Buzz 锁 .....             | 230        |
| 8.1.6        | 信号量 .....                | 231        |
| 8.2          | 内核模块 .....               | 232        |
| 8.2.1        | 模块简介 .....               | 232        |
| 8.2.2        | 装载模块 .....               | 234        |
| 8.2.3        | 卸载模块 .....               | 237        |
| 8.2.4        | 编写内核模块 .....             | 237        |
| 8.3          | 用户内核操作 .....             | 240        |
| 8.3.1        | 编译内核 .....               | 240        |
| 8.3.2        | 系统调用 .....               | 242        |
| 8.4          | 本章小结 .....               | 245        |
| <b>第 9 章</b> | <b>Linux 初始化 .....</b>   | <b>246</b> |
| 9.1          | 开机 .....                 | 246        |
| 9.2          | 系统引导 .....               | 246        |
| 9.3          | Linux 系统初始化 .....        | 248        |
| 9.3.1        | 内核的初始化 .....             | 248        |
| 9.3.2        | init 进程 .....            | 249        |
| 9.4          | 关机 .....                 | 252        |
| 9.5          | 本章小结 .....               | 252        |
| <b>附录 A</b>  | <b>Linux 数据结构 .....</b>  | <b>254</b> |
| <b>附录 B</b>  | <b>GNU 通用公共许可证 .....</b> | <b>296</b> |

# 第 1 章 概 论

## 1.1 初识 Linux

### 1.1.1 Linux 的诞生

Linux 是一个由黑客发展起来的操作系统,它最早是由芬兰人 Linus Torvalds 设计的。当时由于 UNIX 的商业化,Andrew Tannebaum 教授开发了 Minix 操作系统以便于不受 AT&T 许可协议的约束,为教学科研提供一个操作系统。当时 Minix 操作系统被发布在 Internet 上,免费给全世界的学生使用。Minix 具有较多 UNIX 的特点,但与 UNIX 又不完全兼容。1991 年 10 月 5 日,Linus 为了给 Minix 用户设计一个比较有效的 UNIX PC 版本,就动手写了一个“类 Minix”的操作系统,当时最初的内核版本是 0.02。Linus Torvalds 将它发到了 Minix 新闻组,很快就得到了反应。Linus Torvalds 在这种简单的任务切换机制上进行扩展,并在很多热心支持者的帮助下开发和推出了 Linux 的第一个稳定的工作版本。1991 年 11 月 Linux 0.10 版本推出,随后 1991 年 12 月 0.11 版本推出,发布在 Internet 上,免费供人们使用。当 Linux 非常接近于一种可靠的、稳定的系统时,Linus 决定将 0.13 版本称为 0.95 版本。1994 年 3 月,正式的 Linux 1.0 出现了,这差不多就是一种正式的独立宣言了。截止到那时,Linux 用户基数已经发展得很大,而且 Linux 的核心开发队伍也建立起来了。

Linux 是在 Internet 开放环境中开发的,由世界各地的程序员不断完善,而且免费供用户使用。尽管如此,它仍然遵循商业 UNIX 版本的标准。因为前几十年 UNIX 版本的大量出现,电子电气工程协会(IEEE)制定了一个独立的 UNIX 标准,这个新的 ANSI UNIX 标准被称为计算机环境的可移植性操作系统界面(POSIX)。这个标准限定了 UNIX 系统如何进行操作,对系统调用也进行了专门论述。POSIX 限制所有 UNIX 版本必须依赖大众标准,现有大部分 UNIX 和流行版本都是遵循 POSIX 标准的,而 Linux 从一开始就遵循 POSIX 标准。

### 1.1.2 Linux 的发展

Linux 出现后,众多 Linux 爱好者根据自己的使用情况,综合现有的 UNIX 标准和 UNIX 系统中应用程序的特点,不断地修改并增加内容,使 Linux 的功能日益完善。

Linux 提供了与所有主要的窗口管理器相关的接口,以及大量 Internet 工具,如 FTP、TELNET 和 SLIP 等。Linux 提供了比较完整的程序开发工具,最常用的是 C++ 编译器和调试器。

Linux 核心的开发和规范一直是由 Linux 社区控制着,版本也是唯一的。实际上,操作系统的内核版本指的是在 Linus 本人领导下的开发小组开发出的系统内核版本号。自

1994 年 3 月 14 日发布了第一个正式版本 Linux 1.0 以来,每隔一段时间就有新的版本或其修订版公布。发展到今天,Linux 稳定内核的最后版本有:1.0.9、1.2.13、2.0.37、2.2.11,以及最新的 2.4。

在 Linux 的发展历程中还有一项重要的举措:Linux 加入 GNU 并遵循公共版权许可证(GPL)。此举大大加强了 GNU 和 Linux 的应用,几乎所有应用的 GNU 库、软件都移植到 Linux,完善并提高了 Linux 的实用性。在继承自由软件精神的前提下,Linux 不再排斥对自由软件的商业行为,不排斥商家对自由软件进一步开发,不排斥在 Linux 上开发商业软件。从此,Linux 又出现了很多的 Linux 发行版本。目前,最流行的正式版本有:

- Slackware:最早的 Linux 正式版本之一,它遵循 BSD 的风格,尤其在系统启动脚本方面。现有的版本是 Slackware 4.0,它是基于 Linux 2.2 内核的。
- Debian:一个开放源代码的操作系统,目前基于 Linux 2.2 内核。它由许多志愿者维护,是真正的非商业化 Linux,现在较流行的版本是 2.1。
- RedHat Linux:Linux 最早的商业版本之一,在美国和其他英语国家市场上获得了较大成功。现有的最新版本是 RedHat Linux 7.2,基于 Linux 2.4 内核。
- SuSE:它是在欧洲大陆最流行的版本之一,是由德国人开发出来的。现有的最新版本是 SuSE 7.0,基于 Linux 2.2 内核。
- Caldera OpenLinux:最早关注简易安装方法的 Linux 正式版本之一,并在正式版本中集成了办公软件。现有的最新版本是 Caldera OpenLinux 2.2。
- TurboLinux:发行该版本的公司是以推出高性能服务器而著称的 Linux 厂商,在美国有很大的影响。它是占亚洲市场份额最大的商业版本,在中国、日本和韩国都取得了巨大的成功。现在较流行的版本是 TurboLinux 6.0,基于 Linux 2.2 内核。

在中国,Linux 也取得了可喜的发展。从 1999 年 3 月起,国内就涌现出了不少颇具实力的中文 Linux 发行商。例如,中科红旗公司推出的第一个红旗 Linux 桌面版 2.0,采用了多项新技术的蓝点 Linux 2.0,完全符合国际化标准的中文 Linux 发行版本——Turbo Linux 简体中文版 6.0 和 Xteam Linux 3.2 等。

## 1.2 Linux 操作系统结构

### 1.2.1 Linux 的特性

Linux 是一个性能稳定、功能强大、效率高的操作系统。它与 UNIX 系统兼容,在功能特性方面与 UNIX 系统相似,同时又具有自己的独特之处。

#### 1. 多任务

Linux 的多任务性表现在,它可以同时执行多个应用程序,并且不妨碍每个程序的运行。也就是说,用户可以“并行”地处理多个事务。如果用户在 Linux 的一个终端窗口运行程序,在等待过程中,就可以切换到另一个窗口休闲娱乐一番,而这并不妨碍程序的继续执行。

事务处理的“并行”性并不是绝对的。事实上 Linux 使用一种抢占调度的多任务,每

个程序都一直在执行,直到操作系统抢占 CPU 而让其他程序运行时才停止。打开的应用程序看起来像在“并行”运行,只是因为从处理器执行应用程序中的一组指令到 Linux 调度处理器再次运行这个程序之间只有很短的时间延迟。

## 2. 多用户

Linux 支持多个用户同时访问同一台个人计算机上的应用程序。由于 Linux 把微处理器的时间分配给许多应用程序,因此使多个用户可以同时从相同或不同的终端上运行一个或多个应用程序。当多个用户访问同一应用程序时,用户所使用的是程序的副本。

新内核 Linux 2.4 可以处理数量惊人的用户和组——大约 42 亿。

## 3. 多平台

Linux 主要在基于 386/486/586、ISA、EISA、PCI 及 VLB 总线的 PC 上运行。由于 Linux 开放系统的可移植性,使其也可以在 Intel 以外的 CPU 上运行。Linux 是支持硬件平台最多的一种操作系统,其已经支持的硬件平台有:Alpha、Sparc、Arm、M68k、Mips、Ppc 和 S390。内核 2.4 主要的变动是对 IA64 支持,其他新的硬件平台主要是 Super-H 和 Mips64。

Linux 2.4 优越于 Linux 2.2,它能支持最新的 Pentium IV 处理器。MMX 和 MMX2 也被支持,并且也增加了对所有处理器的最优化指令来加速 Linux,尤其对较新的处理器,如 Pentium III 处理器的优化。内核 2.4 中新添加的功能是对 2GB 以上处理器的支持。

## 4. Shell 编程

Linux 允许用户自由地编写脚本程序,这种特性使 Linux 成为一种最灵活的操作系统。Shell 是用户和 Linux 操作系统之间的主要界面,它把在 Linux 环境中建立的许多小命令和许多应用程序连接在一起。用户可以通过编写 Shell 程序来实现许多功能,如使系统个性化、友好化、在后台运行多个进程等。通过 I/O 重定向和后台进程,还可以简化复杂程序。

## 5. 通信和网络

Linux 从一开始就被设计成支持多任务、多用户的操作系统,而且 Linux 的联网能力紧密地与内核结合在一起,这就决定了 Linux 从诞生起就具备强大的联网能力。Linux 支持 TCP/IP 网络协议的 TELNET、FTP、NFS 等服务功能,丝毫不逊色于 Internet。另外,用户也可以通过 Linux 免费获得许多软件,这是通过 Internet 行不通的。

用户通过某些 Linux 命令可以完成内部信息或文件的传输。在外部通信中,Linux 不仅允许传输文件和程序,还允许系统管理员和技术人员访问其他操作系统。

### 1.2.2 Linux 的系统组成

理解 Linux 的系统组成是分析 Linux 内核的基础。Linux 操作系统主要由 4 个子系统组成,从上层到下层依次为:用户进程、O/S 服务、Linux 内核和硬件。其依赖关系表现为:上层依赖下层。

#### 1. 用户进程

用户应用程序是在 Linux 系统上运行的应用程序集合。当某个应用程序在操作系统上运行时,它就成为操作系统的一个进程。

## 2. O/S 服务

O/S 服务是操作系统的一部分,如开窗系统、命令外壳程序、内核编程接口等。

## 3. Linux 内核

这是本文讨论的重点。Linux 内核为用户进程提供了一个虚拟接口,使计算机硬件抽象化。这样用户在对硬件编程处理时就可以使用统一的方式进行。另外, Linux 系统以透明的方式支持多任务,每个进程运行时就好像只有这一个进程在计算机上运行,独占内存和其他硬件资源。而实际上,内核在同时“并行”运行许多进程,并保证各进程访问硬件资源的公平性和进程间的安全性。

Linux 内核主要由 5 个子系统组成。

### 1) 进程调度(SCHED)

进程调度负责控制进程访问 CPU。可运行进程是一个只等待 CPU 资源的进程。Linux 使用基于优先级的简单调度算法来选择下一个运行进程。

### 2) 内存管理(MM)

内存管理子系统是 Linux 操作系统中最重要的组成部分之一。它支持虚拟内存技术,使系统为有限物理内存竞争的进程提供所需的内存空间。

### 3) 虚拟文件系统(VFS)

虚拟文件系统使 Linux 可以支持多个不同的文件系统,每个表示一个虚拟文件系统的通用接口。由于软件将 Linux 文件系统的所有细节进行了转换,所以 Linux 核心的其他部分及系统中运行的程序将看到统一的文件系统。Linux 的虚拟文件系统允许用户同时能透明地安装许多不同的文件系统。

### 4) 网络接口(NET)

网络接口提供了对许多网络标准和网络硬件的访问。

### 5) 进程间通信(IPC)

进程在核心的协调下进行相互间的通信。Linux 支持大量进程间的通信机制。除了信号和管道外, Linux 还支持 UNIX 系统 V(这是 UNIX 的一种变种)中的 IPC 机制。

Linux 内核的结构示意图如图 1-1 所示,从中可以看出各子系统之间的相互依赖关系。

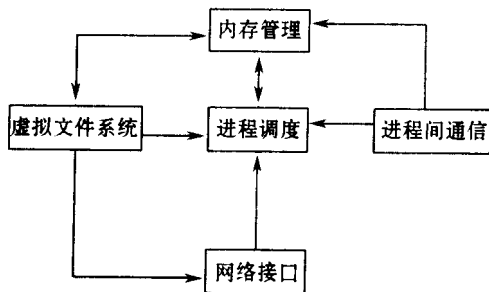


图 1-1 内核子系统

## 4. 硬件

硬件是指安装 Linux 操作系统时所有可能用到的物理设备,如硬盘、内存、主板、CPU 等。

### 1.2.3 Linux 与其他操作系统的区别

Linux 可以与 MS-DOS、OS/2、Windows 等其他操作系统共存于同一台机器上,但 Linux 是开放、免费的操作系统,而其他操作系统都是封闭系统,需要有偿使用。

#### 1. Linux 与 MS-DOS 的区别

就操作系统的功能而言,MS-DOS 是单任务的操作系统。一旦用户运行了一个 MS-DOS 的应用程序,它就独占了系统的资源,用户也就不可能再同时运行其他应用程序。而 Linux 是多任务的操作系统,用户可以同时运行多个应用程序。

就发挥处理器功能而言,MS-DOS 没有完全实现 x86 处理器的功能。而 Linux 完全在处理器保护模式下运行,并且发挥了处理器的所有特性。Linux 提供了完整的 UNIX 接口,可以直接访问计算机的所有可用内存。而 MS-DOS 只支持部分 UNIX 接口。

#### 2. Linux 与 OS/2、Windows、Windows NT 之间的区别

Linux 是从一个比较成熟的操作系统发展而来的,而其他操作系统如 Windows NT 等都是自成体系的,没有对应的相依托的操作系统。这一区别使 Linux 用户能从 UNIX 团体贡献中获利。UNIX 是世界上使用最普遍、发展最成熟的操作系统之一。由于许多计算机系统(从个人计算机到超级计算机)都存在 UNIX 版本,所以 UNIX 用户就可以从很多方面得到支持和帮助。因此,Linux 作为 UNIX 的一个变体,同样会得到相应的支持和帮助。

## 1.3 Linux 的内核源代码

### 1.3.1 如何得到源代码

Linux 的内核源代码可以从很多途径得到。一般来讲,在 Linux 系统根目录下,usr/src/linux 目录下的东西就是内核源代码。源码的版本号有一个非常简单的编号约定:任何偶数的内核(如 2.0.30)都是一个稳定发行的内核,而任何奇数的内核(如 2.1.42)都是一个开发中的内核。另外,源码还可以从互联网上下载,解压缩后文件一般也都位于 Linux 目录下。内核源代码有很多版本,目前最新的稳定版本是 2.4。

### 1.3.2 源代码的目录体系

内核源程序的文件按树形结构进行组织,在源程序树最上层的用户可以看到如下一些目录。

#### 1) arch

arch 子目录包括了所有与体系结构相关的内核代码。它的每一个子目录都代表一种支持的体系结构。例如,i386 就是关于 intel CPU 及与之相兼容的体系结构的子目录。PC 机一般都基于此目录。

#### 2) include

include 子目录包括编译内核所需要的大部分头文件。与平台无关的头文件在 include/linux 子目录下,与 intel CPU 相关的头文件在 include/asm-i386 子目录下,而 in-

clude/scsi 目录则是有关 SCSI 设备的头文件目录。

### 3) init

此目录不但包含内核的初始化代码(不是系统的引导代码),而且包含 main.c 和 Version.c 文件。这是一个非常好的研究内核如何工作的起点。

### 4) mm

此目录包括所有独立于 CPU 体系结构的内存管理代码,如页式存储管理内存的分配和释放等。而与体系结构相关的内存管理代码则位于 arch/\* /mm/中,如 arch/i386/mm/Fault.c。

### 5) kernel

此目录包括主要的内核代码,此目录下的文件实现了大多数 Linux 系统的内核函数,其中最重要的文件是 sched.c。与体系结构相关的代码在 arch/\* /kernel 中。

### 6) drivers

此目录用于放置系统所有的设备驱动程序;每种驱动程序各占用一个子目录:在/block 目录下的程序(如 ide)为块设备驱动程序。如果用户希望查看所有可能包含文件系统的设备是如何初始化的,就可以查看 drivers/block/genhd.c 中的 device\_setup() 函数。它不仅初始化硬盘,也初始化网络,因为安装 NFS 文件系统需要网络。

### 7) 其他

lib 目录放置内核的库代码;net 目录包含内核与网络相关的代码;ipc 目录包含内核进程间通信的代码;fs 目录是所有的文件系统代码和各种类型的文件操作代码,它的每一个子目录支持一个文件系统,如 FAT 和 EXT2;scripts 目录包含用于配置内核的脚本文件等。每个目录下一般都有一个 depend 文件和一个 makefile 文件,这两个文件都是编译时使用的辅助文件,仔细阅读这两个文件对弄清各个文件之间的相互依托关系很有帮助。有的目录下还有 readme 文件,它是对相应目录下的文件的一些说明,同样有利于用户对内核源码的理解。

## 1.3.3 源代码的解读

俗话说:“工欲善其事,必先利其器”。Linux 内核是一个庞大而复杂的系统,其源代码之间通常会交叉引用。所以在解读过程中,读者为弄清内核源码的某一部分往往要查阅内核源码的另一部分,而这一部分通常会涉及许多其他相关文件,以至于最后很难回到原来的起点。源代码导航器的出现解决了这一问题,相关站点为:<http://lxr.linux.no/source>。

### 1. 系统启动与初始化

在基于 Intel 的系统中,由 loadlin.exe 或 LILO 将内核载入内存并启动内核,之后控制权被转移给内核。这一部分的初始化位于 arch/i386/kernel/head.s 中。head.s 在设置一些特定体系结构后会跳转到 init/main.c 的 main() 例程中。



**注意:**如果文中没有特殊说明,所有的文件引用都是相对于/usr/src/linux-2.4 的。

### 2. 内存管理

内存管理的源代码主要在 mm 目录中,但与特定体系结构相关的部分则在 arch/\* /mm 中;内存缺页处理代码在 mm/memory 中;内存映射和页缓冲代码在 mm/filemap.c

中;缓冲区缓存代码在 `mm/buffer.c` 中;页交换代码在 `mm/swap_state.c` 和 `mm/swapfile.c` 中。

### 3. 内核

`kernel` 中包含大部分内核代码。与特定体系结构相关的部分在 `arch/*/kernel` 中;调度进程代码在 `kernel/sched.c` 中;fork 代码在 `kernel/fork.c` 中;Bottom Half 控制程序代码在 `include/linux/interrupt.h` 中;task\_struct 数据结构在 `include/linux/sched.h` 中。

### 4. PCI

PCI 伪驱动程序在 `drivers/pci/pci.c` 中,定义在 `include/linux/pci.h` 中。每种体系结构都有其特定的 PCI BIOS 代码。

### 5. 进程间通信

`ipc` 目录包含大部分的相关代码,所有的 System V IPC 对象都有一个定义在 `include/linux/ipc.h` 中的 `ipc_perm` 数据结构。System V 消息机制代码在 `ipc/msg.c` 中,共享内存代码在 `ipc/shm.c` 中,信号量代码在 `ipc/sem.c` 中,管道代码在 `ipc/pipe.c` 中。

### 6. 中断处理

内核的中断处理代码几乎都与特定的微处理器相关。Intel 中断处理代码在 `arch/i386/kernel/irq.c` 中,并定义在 `include/asm_i386/irq.h` 中。

### 7. 设备驱动程序

`drivers` 中包含 Linux 所有的设备驱动程序代码。根据类型可分为如下几种。

- `/block`: 块设备驱动程序。块设备包含 IDE 和 SCSI 设备。其初始化过程参见 `drivers/block/genhd.c` 中的 `device_setup()` 函数,该函数不仅可以初始化硬盘,也可以初始化网络。
- `/char`: 字符设备驱动程序,如 `ttys`、串口和鼠标等。
- `/cdrom`: 包含 Linux 所有 CD-ROM 的代码。需要注意的是,IDE CD 驱动程序在 `drivers/block/ide-cd.c` 中,而 SCSI CD 驱动程序在 `drivers/scsi/scsi.c` 中。
- `/pci`: 即 PCI 伪驱动程序代码,包含 PCI 子系统映射和初始化原理。
- `/scsi`: 包含所有的 SCSI 代码以及 Linux 所支持的所有 SCSI 设备驱动程序代码。
- `/net`: 网络设备驱动程序。
- `/sound`: 声卡设备驱动程序。

### 8. 文件系统

EXT2 文件系统代码在 `fs/ext2/` 目录下,其数据结构定义在 `include/linux/ext2_fs.h`、`ext2_fs_i.h` 和 `ext2_fs_sb.h` 中。虚拟文件系统代码在 `fs/*` 中,其数据结构定义在 `include/linux/fs.h` 中。缓冲区缓存代码在 `fs/buffer.c` 中。

### 9. 网络

网络部分代码在 `net` 中,其大部分的 `include` 文件在 `include/net` 中。BSD 套接字代码在 `net/socket.c` 中,IP 版本 4 INET 套接字代码在 `net/ipv4/af_inet.c` 中,常用的协议支持代码(包括 `sk_buff` 控制例程)在 `net/core` 中,TCP/IP 网络代码在 `net/ipv4` 中,而网络设备驱动程序在 `drivers/net` 中。

### 10. 模块

内核模块代码一部分在内核中,一部分在模块包中。内核代码在 `kernel/modules.c`