

Broadview®
www.broadview.com.cn

深入浅出 Linux 工具与编程

余国平◎著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

深入浅出 Linux 工具与编程

余国平◎著

- ◎ 这是一本有思想、有内容的图书
- ◎ 这是一本回答学什么、怎么学的图书
- ◎ 这是一本大学毕业生一次阅读、终生受益的图书
- ◎ 这是一本集实用性、典型性、模仿性案例的图书
- ◎ 这是一本很通俗易懂的图书
- ◎ 这是一本帮你突破技术玻璃纸、一通百通的图书
- ◎ 这是一本包含作者从业十年心得经验的图书
- ◎ 这是一本零起点的Linux专家速成培训教程

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书系统地论述了 Linux 工具与编程的相关知识。全书内容可分为两部分：Linux 知识的初级部分和高级部分。其中初级部分包括 Linux 操作系统介绍、Linux 命令说明、Linux 常见实用工具（正则表达式、find、sed、awk）、Shell 编程、Linux C 语言程序设计、Linux C 语言开发工具（vi 与 vim 编辑器、gcc、Makefile 和 gdb）；高级部分包括 Linux 进程编程（Linux 进程、Linux 线程、管道与信号、消息队列、信号量和共享内存）、Linux 文件编程、网络编程和 XML 编程。

本书初级部分适合高等院校相关专业学生和 Linux 爱好者学习，高级部分适合 Linux 行业资深从业人员学习。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

深入浅出 Linux 工具与编程 / 余国平著. —北京：电子工业出版社，2011.7
ISBN 978-7-121-13750-1

I. ①深… II. ①余… III. ①Linux 操作系统 IV. ①TP316.89

中国版本图书馆 CIP 数据核字（2011）第 104466 号

策划编辑：张春雨

责任编辑：李利健 贾 莉

印 刷：北京天宇星印刷厂

装 订：三河市皇庄路通装订厂

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1092 1/16 印张：35.5 字数：908 千字

印 次：2011 年 7 月第 1 次印刷

印 数：4000 册 定价：79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前言

作者在软件行业从业了十年，先后通过了国家软件水平等级考试的《高级程序员》级别和《系统分析师》级别，参加了大型行业软件如银行核心业务系统、前置系统、数据仓库、金卡工程、银行大小额现代化支付、中间件、支付宝银行端接口等一系列项目设计、开发、测试和实施工作，具有丰富的软件编程经验，同时，一直在多家企业负责新员工的培训工 作，对企业员工培训有较多的心得。新员工在大量繁多的技术面前常碰到应该学什么、怎么学的问题；各种技术与工具知识点怎么分优先级和重点问题；技术玻璃纸难以突破，水平难以提高的问题。在企业里，员工怎样在时间有限、精力有限的情况下掌握好实用和有用的技术，满足企业用人的需要？作者总结的经验是培训教材的导向必须注重目的性、思想性、实践性、典型性和实用性，以任务驱动式培训和以目标管理为方法，用图文和言简意赅的语言描述技术思想，用经典程序说明技术思想，用多个项目实战案例解释如何高水平运用技术思想。

模仿是人们快速提高自身能力的捷径，本书的许多程序十分经典，采用实际编程使用的架构，读者可以将编程模板应用于实践。本书的编排注重易学习性、可模仿性和实战性，其中模板化的编程案例和规范化的练习可以让读者在短时间内把书本知识变成自身的能力。本书是一本技术思想深厚的图书，书中的许多内容来源于作者十年技术积累的总结，本书有些技术概念的概括来源于作者多年的思考和感悟，项目案例来源于作者从业的实际项目中。

本书由长期从事一线开发的技术人员编写完成，书中内容通俗易懂，作者力求让 Linux 技术变得简单，读者在阅读本书后能大大增加学习 Linux 技术的信心。本书把繁多的 Linux 技术进行了浓缩，能大大节约读者的学习时间和学习成本。本书注重对技术概念的简要阐述，更注重技术实现，书中对一些技术细节的归纳来源于作者多年工作经验的总结。没有理论，实践是盲目的；没有实践，理论是空洞的。本书力求用言简意赅的理论让读者掌握技术的精髓，用经典程序和项目案例使读者加深对技术理论的理解。本书用精练的概念总结技术，用通俗易懂的语言说明技术，用精心挑选的模板程序和项目案例实现技术。

本书内容

本书分 6 篇，所有的内容注重理论联系实际。每篇的主要内容如下：

第 1 篇 Linux 命令及其工具

本篇包括 Linux 操作系统介绍、Linux 命令说明、Linux 常见实用工具(正则表达式、find、sed、awk) 说明及实例练习、Shell 编程语法说明及编程实例。

第 2 篇 Linux C 语言程序设计

本篇包括 C 语言基础、C 语言控制结构、C 语言函数、C 语言数组、结构体及指针、C 语言预编译、格式化 I/O 函数、字符串和内存操作函数、字符类型测试函数、字符串转换函数、Linux C 语言开发工具 (vi 与 vim 编辑器、gcc、Makefile 和 gdb)。本篇多次运用堆栈表格对程序运行进行解释，这对于理解计算机语言运行机理非常重要。只有理解的才是最深刻的，理解其运行机理，可以触类旁通、一通百通，移植到理解 C++ 语言和 Java 等语言。

第 3 篇 Linux 进程

本篇包括 Linux 编程基本概念、Linux 进程、Linux 线程、管道与信号、消息队列、信号量和共享内存。Linux 进程章节中守护进程模板和数据仓库多进程处理案例可以应用到实际项目中。本篇 Linux 进程间通信程序范例是实际项目中精简的 Demo 程序，程序模型和使用方法与实际项目中类似。

第 4 篇 Linux 文件

本篇包括 Linux 文件编程，该部分内容对文件函数进行了分类总结，并提供了典型范例。

第 5 篇 网络编程

本篇包括网络知识基础、Socket 编程。Socket 编程章节包括 TCP 并发服务器案例、TCP 迭代服务器案例、文件服务器案例、UDP 服务器编程、UDP 广播、UDP 多播、UNIX/Linux 域套接字编程等。

第 6 篇 XML 编程

本篇包括 XML 概念、XML 语法、XPath 语法、libxml 编程、支付宝银行端接口 XML 项目案例。本篇内容是目前市面上对 Linux 下 XML 编程总结非常全面的，在实际项目开发中有较大的借鉴意义。

本书特色

1. 零起点的企业级培训教程

读者只需具有大学计算机专业及相关专业一般水平，即可对本书进行阅读和练习。本书内容

通俗易懂，图文并茂，注重知识点的总结概括和分类。知识结构注重层层递进，以达到让读者在低起点向专家迈进的目的。有 Linux 从业基础部分（如 Linux C 语言程序设计），有 Linux 从业素质能力培养部分（如 Linux 命令及其工具、Linux C 语言开发工具），也有 Linux 编程专家水平能力训练部分（如 Linux 进程编程、Linux 文件编程、Linux 进程间通信、网络编程与 XML 编程）。本书涵盖了 Linux 原理篇、命令篇、工具篇和程序篇。

专业就是“简单的事情”重复做，做到专业，就是把复杂的事情简单化，其方法为分类、分层、总结、模板化和流程化。而本书正是致力于这一目的，把复杂的技术简单明了地呈现在读者面前，帮助读者成为专业人士。

2. 大量的企业级实训内容

本书的许多章节是作者关于企业级培训的实训内容，知识点注重目标明确、言简意赅、分清主次、项目导向，以求达到简洁不简单效果。本书属于企业级实训教程，以 Linux 行业从业素质能力培养为导向，以实际应用为目标，以简洁的理论和经典练习为过程，以期达到快速提高读者的职业水平和职业能力。本书采用 Linux 行业素质能力模型的训练方法，即将 Linux 从业知识点逐条列出，并把知识点整合到规范化练习案例中，以达到让读者通过模仿练习快速把知识变为能力的目的。如本书 Linux 工具与命令章节，读者只要按照练习，即可达到 Linux 行业从业所需的中级水平，而 Shell 章节按照练习即可快速提高到 Linux 行业老员工的水平。本书这些企业级培训内容能帮助读者在短时间内学到实用且够用的 Linux 开发知识。

3. 学什么，怎么学

一门技术是很难在短时间内学好的，但通常可以快速学会常用和关键的技术。本书以实用论为导向，丢弃了项目开发中用不到的众多技术细节。

本书注重理论联系实际，作者把自己十年的项目经验整合到本书中，将 Linux 从业的知识进行分类、总结，并辅以案例讲解，许多知识点都以实际工作所需知识为准，也是以作者所掌握的主要和重点知识为准。

书中的许多章节配有典型程序和规范化案例练习，学完理论后按照案例练习，即可达到技术的提升。本书内容的选取完全参照作者十年从业经验所用到的知识，言简意赅的图文讲解和规范化案例练习告诉读者怎么学。

4. 多个实用项目案例

本书包含多个经典的项目案例，如 Shell 章节的备份脚本、C 语言章节的实用日志库、Linux 进程章节数据仓库多进程案例、网络章节的实用文件服务器和实用通信库、XML 章节支付宝银行端接口项目的 XPath 库。这些案例具有较大的实用参考价值。

5. XML 章节填补市场空白

XML 是软件行业经常使用的技术，经常应用在数据交换、Web 服务、内容管理、电子商务、配置脚本等方面。目前市面上的图书缺少针对 XML 开发技术的案例介绍，本书对这部分内容进行了专门的总结，同时提供了丰富的练习和经典的项目案例。

6. Linux 专家速成培训教程

时间是人类发展的空间，赢得时间就是赢得个人发展的空间。在个人的职业生涯中，一步领先常常可以做到步步领先。读者只要静下心来用一个月时间阅读本书，并进行练习，就可以大大提高 Linux 编程技术水平，如果完全掌握本书内容，即可达到 Linux 专家水平。可以说，本书是一本通向 Linux 专家之路的速成教程。

由于作者水平有限，书中错漏之处在所难免，恳请读者批评指正。

目录

第 1 篇 Linux 命令及其工具	
第 1 章 Linux 系统与命令2	
1.1 Linux 操作系统..... 2	
1.1.1 Linux 重要概念.....2	
1.1.2 Linux 组成.....3	
1.1.3 Linux 目录结构.....3	
1.1.4 Linux 操作系统的组成.....5	
1.1.5 Linux 用户管理.....6	
1.1.6 Linux 文件管理.....6	
1.2 Linux 命令..... 7	
1.2.1 Linux 命令帮助.....7	
1.2.2 Linux 命令的符号及意义.....8	
1.2.3 Linux 命令.....9	
第 2 章 Linux 常用实用工具 18	
2.1 正则表达式.....18	
2.2 find 查找命令.....20	
2.2.1 find 语法.....20	
2.2.2 find 实例练习.....22	
2.3 sed.....24	
2.3.1 sed 语法.....24	
2.3.2 sed 实例练习.....26	
2.4 awk.....29	
2.4.1 awk 语法.....29	
2.4.2 awk 实例练习.....36	
第 3 章 Shell 编程 40	
3.1 Shell 环境变量..... 40	
3.1.1 环境变量说明.....40	
3.1.2 用户常用的系统环境变量.....41	
3.1.3 用户登录脚本示例.....42	
3.2 Shell 的符号、变量及运行..... 43	
3.2.1 Shell 中的符号及其含义.....43	
3.2.2 “反引号命令替换”.....44	
3.2.3 Shell 变量.....45	
3.2.4 Shell 脚本执行.....50	
3.2.5 Shell 退出状态.....50	
3.3 Shell 的输入和输出..... 51	
3.3.1 Shell 的输入.....51	
3.3.2 Shell 的输出.....52	
3.4 Shell 测试条件..... 53	
3.5 Shell 的流程控制结构..... 57	
3.5.1 if 语句.....57	
3.5.2 case 语句.....59	
3.5.3 while 语句.....60	
3.5.4 until 语句.....61	
3.5.5 for 语句.....62	
3.5.6 跳转语句.....64	
3.6 Shell 数组..... 64	

3.7	Shell 函数	65	6.3.1	指针概念	134
3.8	I/O 重定向	67	6.3.2	sizeof、void、const 说明	139
3.9	Shell 内置命令	68	6.3.3	指针变量作为函数参数	140
3.10	实用 Shell 脚本	73	6.3.4	指针的运算	142
第 2 篇 Linux C 语言程序设计					
第 4 章 C 语言基础 76					
4.1	C 语言基本概念	76	6.3.5	指向数组的指针变量	144
4.2	常量与变量	84	6.3.6	数组名作为函数参数	146
4.3	运算符	89	6.3.7	函数指针变量	148
4.4	C 语言控制结构	96	6.3.8	返回指针类型函数	149
4.4.1	if 语句	97	6.3.9	指向指针的指针	150
4.4.2	switch 语句	100	6.3.10	结构指针	150
4.4.3	goto 语句	101	6.3.11	动态存储分配	152
4.4.4	while 语句	102	6.3.12	指针链表	153
4.4.5	do-while 语句	103	6.3.13	指针数据类型小结	154
4.4.6	for 语句	104	第 7 章 C 语言预处理 156		
4.4.7	break 和 continue 语句	106	7.1	define 宏定义	156
第 5 章 C 语言函数 107					
5.1	函数简述	107	7.2	typedef 重定义	157
5.2	函数变量	110	7.3	inline 关键字	158
5.3	函数定义与调用	110	7.4	条件编译	158
5.3.1	函数定义	110	7.5	头文件的使用	159
5.3.2	函数的参数与返回值	111	第 8 章 格式化 I/O 函数 161		
5.3.3	函数调用	115	8.1	格式化输出函数	161
第 6 章 C 语言数组、结构体及指针 119					
6.1	C 语言数组	119	8.1.1	输出函数原型	161
6.1.1	数组概述	119	8.1.2	输出函数格式说明	162
6.1.2	一维数组	120	8.2	格式化输入函数	165
6.1.3	二维数组	124	8.2.1	输入函数原型	165
6.1.4	字符数组	127	8.2.2	输入函数格式说明	165
6.1.5	冒泡法排序	128	第 9 章 字符串和内存操作函数 169		
6.2	C 语言结构	129	9.1	字符串操作函数说明	169
6.2.1	结构概念	129	9.2	字符串函数操作	170
6.2.2	结构变量	131	9.3	字符类型测试函数	179
6.3	指针	134	9.4	字符串转换函数	180
第 10 章 标准 I/O 文件编程 182					
10.1 文件打开方式 183					
10.2 标准 I/O 函数说明及程序范例 185					
第 11 章 Linux C 语言开发工具 200					
11.1 vi 与 vim 200					

11.1.1	vi 与 vim 概述	200	第 13 章	Linux 线程编程	300
11.1.2	指令模式	201	13.1	线程简要说明	300
11.1.3	末行模式	207	13.2	线程主要函数	302
11.1.4	vim 个人使用经验	210	13.3	线程编程	308
11.1.5	vim 的使用	211	13.3.1	线程创建	308
11.1.6	文件编码	214	13.3.2	终止线程	310
11.1.7	vi 与 vim 模拟练习	217	13.3.3	线程互斥	312
11.2	gcc	218	13.3.4	线程同步	315
11.2.1	gcc 简要说明	218	第 14 章	Linux 进程间通信——管道与信号	318
11.2.2	gcc 参数	220	14.1	进程间通信概述	318
11.3	Makefile	224	14.2	管道	319
11.3.1	Makefile 简介	225	14.2.1	pipe 管道	320
11.3.2	Makefile 语法	227	14.2.2	标准流管道	324
11.3.3	Makefile 的运行	231	14.2.3	命名管道 (FIFO)	325
11.3.4	Makefile 的扩展话题	232	14.3	信号	328
11.4	gdb	233	14.3.1	信号概述	328
11.4.1	gdb 语法	233	14.3.2	信号的发送和捕捉函数	332
11.4.2	gdb 调试	238	14.3.3	信号的处理	337
第 3 篇 Linux 进程					
第 12 章	Linux 进程编程	242	第 15 章	System V 进程间通信	346
12.1	Linux 进程编程基本概念	242	15.1	System V 进程间通信的键值	346
12.1.1	登录	242	15.2	消息队列	350
12.1.2	文件和目录	243	15.2.1	消息队列简要说明	351
12.1.3	输入和输出	243	15.2.2	消息队列函数	352
12.1.4	程序与进程	244	15.2.3	消息队列使用程序范例	355
12.1.5	ANSI C	245	15.3	信号量	360
12.1.6	用户标识	247	15.3.1	信号量简要说明	360
12.1.7	出错处理	247	15.3.2	信号量函数	361
12.1.8	Linux 信号、时间值与系统调用	249	15.3.3	信号量应用程序示例	364
12.2	Linux 进程环境	256	15.4	共享内存	366
12.3	Linux 进程控制	267	15.4.1	共享内存简要说明	366
12.4	进程关系	289	15.4.2	共享内存函数	368
12.5	守护进程与多进程并发案例	293	15.4.3	共享内存应用范例	370
12.5.1	守护进程的编写	293	第 4 篇 Linux 文件		
12.5.2	多进程并发项目案例	296	第 16 章	Linux 文件编程	376
			16.1	文件系统函数	376

16.2	初级文件 I/O 函数	392	18.6	本地进程间套接字	478
16.3	标准 I/O 的缓冲和刷新	399	18.6.1	非命名 UNIX 域套接字管道	478
第 5 篇 网络编程					
第 17 章	网络知识基础	402	18.6.2	UNIX 域套接字	479
17.1	网络体系结构及协议	402	18.7	I/O 编程模型	483
17.1.1	网络体系结构概念	402	第 6 篇 XML 编程		
17.1.2	TCP/IP 模型	405	第 19 章	XML 概念与语法	490
17.1.3	网络分类与广域网	407	19.1	XML 概述	490
17.1.4	网络地址	410	19.2	XML 语法	493
17.2	TCP/IP 协议簇报文格式	412	19.3	XPath 语法	503
第 18 章	socket 编程	416	19.3.1	XPath 基本语法	503
18.1	套接字说明及函数说明	416	19.3.2	XPath 位置路径	508
18.1.1	套接字说明	416	19.3.3	XPath 示例	514
18.1.2	socket 地址说明及 转换函数	419	第 20 章	libxml 编程	518
18.1.3	socket 主要函数说明	424	20.1	libxml 编程基础	518
18.2	TCP 套接字编程	432	20.1.1	libxml 的安装	518
18.2.1	TCP 套接字编程模型	432	20.1.2	libxml 主要的数据类型	519
18.2.2	迭代服务器编程	436	20.1.3	libxml 的主要函数说明	522
18.2.3	并发服务器编程	437	20.1.4	XML 常见操作	530
18.3	TCP 文件服务器项目案例	443	20.2	libxml 高级编程进阶	536
18.4	UDP 编程	458	20.2.1	理解 DOM 树	536
18.4.1	普通 UDP 服务器编程	458	20.2.2	libxml 编程实例练习	541
18.4.2	UDP 广播	461	20.2.3	支付宝银行端接口 XML 项目案例	547
18.4.3	UDP 多播	464	附录		552
18.5	原始套接字	469	参考文献		555
18.5.1	原始套接字说明	469			
18.5.2	原始套接字举例	471			

第 1 篇

Linux 命令及其工具

- ❖ 第 1 章 Linux 系统与命令
- ❖ 第 2 章 Linux 常用实用工具（正则表达式、find、sed、awk）
- ❖ 第 3 章 Shell 编程

学海聆听：

- 合抱之木，生于毫末；九层之台，起于垒土；千里之行，始于足下。
- 学而不思则罔，思而不学则殆。
- 知识来源于学习、思考、顿悟、实践和交流。
- 非学无以广才，非志无以成学。
- 临渊羡鱼，不如退而结网。
- 纸上得来终觉浅，绝知此事要躬行。
- 在模仿中成长，在创新中成功。
- 好的开始等于成功的一半。
- 梦想引领未来，行动成就精彩。
- 确定目标—支配行动—克服困难—实现目标。
- 工欲善其事，必先利其器。
- 博学之，审问之，慎思之，明辨之，笃行之。

第 1 章

Linux系统与命令

本章分两部分：Linux 操作系统和 Linux 命令。Linux 操作系统部分简要介绍 Linux 操作系统的概念、组成和实现。Linux 命令部分用表格列出了 Linux 常见命令和 Linux 次常见命令，简要地说明了其使用方法。熟练地掌握 Linux 命令和工具是非常必要的，能提高工作效率。

1.1 Linux 操作系统

1.1.1 Linux 重要概念

Linux 操作系统有如下一些重要的概念。

① 机器指令：CPU 可识别的二进制指令集，许多指令可以与汇编指令一一对应。主要分为四类：计算指令、存数据指令、取数据指令、中断指令。

② 程序：用计算机语言编写的源代码叫做源程序，经过编译程序编译后形成的可执行码叫做可执行程序，程序是静态的。

③ 进程：程序的一次执行过程叫做进程。进程是动态的，其堆栈空间存在于内存中，CPU 从内存中取程序执行代码执行相关操作。一个程序执行一次，就产生一个进程，所以，程序与进程是一对多的关系。

④ 中断：顾名思义，即中止和打断。中断是操作系统中止当前运行程序去执行相应中断号程序的过程，执行完成中断号程序后，回到原程序断点并往下继续执行。

⑤ 文件系统：组织、存储、检索和管理文件的系统。

⑥ 文件：一个具有符号的一组相关联元素的有序序列。

⑦ 文件名：即文件的名称，一个目录下的文件名不能重复。

⑧ 目录：为了方便管理文件，就形成了树形目录结构。目录下可以有文件或目录，目录类似于 Windows 系统下的文件夹。

⑨ 相对路径：相对当前路径的路径。如当前目录为 `/home/test/hello`，到达 `/home/`

test/hello/why 目录，可通过相对路径方式 `cd ./why` 到达。

⑩ 绝对路径：从根目录 (/) 开始的路径。如当前目录为 /home/test/hello，若要到达 /home/test/hello/why 目录，可通过绝对路径方式 `cd /home/test/hello/why` 实现。

⑪ Shell：Shell 是一个交互性命令解释器，Linux 用户可以通过编写 Shell 程序完成一些管理工作。

⑫ inode 节点：文件系统中的基本对象。每个文件和目录对应一个 inode，inode 在一个分区内是唯一的，它可以是一个正常文件、一个目录、一个符号链接，或者是其他内容。

1.1.2 Linux 组成

Linux 有四个主要部分，分别为 Shell、文件系统、内核、实用工具。

Linux 内核主要由五个子系统组成，分别为进程调度、内存管理、虚拟文件系统、网络接口、进程间通信。

Linux 系统管理包括用户管理、文件管理、设备管理、进程管理、网络管理、作业管理和实用工具。Linux 实用工具包括压缩/解压工具、Shell 工具等。

1.1.3 Linux 目录结构

1. 目录结构

Linux 文件系统是树状目录层次结构。/ 为根目录，根目录为一级子目录，各子目录名有特定的含义，这些目录存放和管理特定的文件。Linux 目录结构如图 1-1 所示。

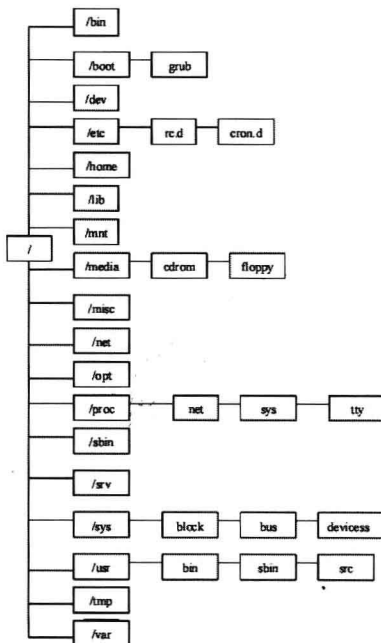


图 1-1 Linux 目录结构

2. 目录说明

为了方便文件的管理，Linux 操作系统建有许多一级子目录或多级子目录，这些目录存放和管理特定的某一类文件。Linux 目录结构说明如表 1-1 所示。

表 1-1 Linux 目录结构说明

目录名称	说 明
/bin	bin 就是二进制 (binary) 英文缩写。该目录下存放 Linux 常用操作命令的执行文件，如 mv、ls、mkdir 等。有时，这个目录的内容和 /usr/bin 中的内容一样，它们都是放置一般用户使用的执行文件
/boot	这个目录下存放着操作系统启动时所要用的程序
/dev	该目录包含了 Linux 系统中使用的所有外部设备。要注意的是，这里存放的并不是外部设备的驱动程序，它实际上是一个访问这些外部设备的端口。由于在 Linux 中，所有的设备都当做文件进行操作，比如：/dev/cdrom 代表光驱，用户可以非常方便地像访问文件、目录一样对其进行访问
/etc	该目录下存放了系统管理时要用到的各种配置文件和子目录，如网络配置文件、文件系统、系统配置文件、设备配置信息设置用户信息等。系统在启动过程中需要读取其参数进行相应的配置
/etc/rc.d	该目录主要存放 Linux 启动和关闭时要用到的脚本文件
/etc/rc.d/init	该目录存放 Linux 服务默认的所有启动脚本。在新版本的 Linux 中还用到 /etc/xinetd.d 目录下的内容
/home	该目录是 Linux 系统中默认的用户工作根目录。如果建立一个名为“xx”的用户，那么在 /home 目录下就有一个对应的“/home/xx”路径用来存放该用户的主目录
/lib	该目录是用来存放系统动态链接共享库的。几乎所有的应用程序都会用到这个目录下的共享库。因此，千万不要轻易对这个目录进行任何操作
/lost+found	该目录在大多数情况下都是空的。只有当系统产生异常时，才会将一些遗失的片段放在此目录下
/media	该目录下以前是光驱和软驱的挂载点，现在 Linux 已经可以自动挂载光驱和软驱了
/misc	该目录下存放从 DOS 下进行安装的实用工具，一般为空
/mnt	该目录是软驱、光驱、硬盘的挂载点，也可以将别的文件系统临时挂载到此目录下
/proc	该目录是用于放置系统核心与执行程序所需的一些信息，这些信息是在内存中由系统产生的，故不占用硬盘空间
/root	该目录是超级用户登录时的主目录
/sbin	该目录用来存放系统管理员常用的系统管理程序
/tmp	该目录用来存放不同程序执行时产生的临时文件。Linux 安装软件的默认安装路径通常位于该目录下
/usr	这是一个非常重要的目录，用户的很多应用程序和文件都存放在这个目录下，类似于 Windows 下的 Program Files 目录
/usr/bin	系统用户使用的应用程序
/usr/sbin	超级用户使用的比较高级的管理程序和系统守护程序
/usr/src	内核源代码默认的放置目录
/srv	该目录存放一些服务启动之后需要提取的数据
/sys	这是 Linux 2.6 内核的一个很大的变化。该目录下安装了 Linux 2.6 内核中新出现的一个文件系统-sysfs 文件系统，此系统集成了三种文件系统的信息：针对进程信息的 proc 文件系统、针对设备的 devfs 文件系统以及针对伪终端的 devpts 文件系统。/sys 目录是内核设备树的一个直观反映。当一个内核对象被创建的时候，对应的文件和目录也在内核对象子系统中被创建
/var	这也是一个非常重要的目录，很多服务的日志信息都存放在这里

1.1.4 Linux 操作系统的组成

Linux 操作系统的层次结构如图 1-2 所示。用户应用程序如办公软件，或者自己编写的应用程序服务；操作系统服务指 Shell 解释器、系统调用接口等；操作系统内核是操作系统在内核态完成系统功能的程序；硬件系统指硬件设备驱动程序。

Linux 内核系统模块主要由五个模块构成，它们分别是：进程调度模块、内存管理模块、文件系统模块、进程间通信模块和网络接口模块。它们的相互依赖关系如图 1-3 所示，由图可以看出，所有的模块都与进程调度模块存在依赖关系，因为它们都需要依靠进程调度程序来挂起（暂停）或重新运行它们的进程。通常，一个模块会在等待硬件操作期间被挂起，而在操作系统完成硬件操作后才可继续运行。

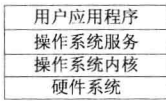


图 1-2 Linux 操作系统的层次结构图

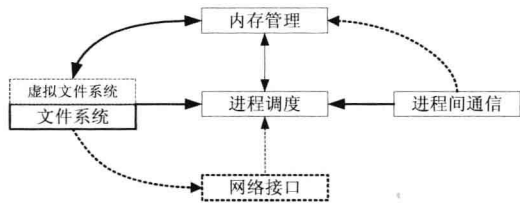


图 1-3 Linux 内核系统模块结构及相互依赖关系图

Linux 操作系统属于系统软件，软件都是建立在硬件基础上去实现相应的功能。Linux 内核结构图如 1-4 所示，此图划分了三个层次，分别为硬件级、内核级、用户级。硬件级指的是电脑硬件，如 CPU、显卡、显示器等。内核级指的是操作系统内核功能（除硬件控制部分），它处于操作系统的核心态，内核代码可以无限制地对系统存储、外部设备进行访问。图中内核级的硬件控制一般指的是 BIOS（基本输入/输出系统）程序，BIOS 的主要功能是为计算机提供最底层的、最直接的硬件设置和控制。用户级是操作系统提供给用户的调用接口，它处于操作系统的用户态，应用程序要访问系统硬件资源需要通过系统调用去实现，应用程序调用系统调用时，用户进程将由用户态切换到核心态。

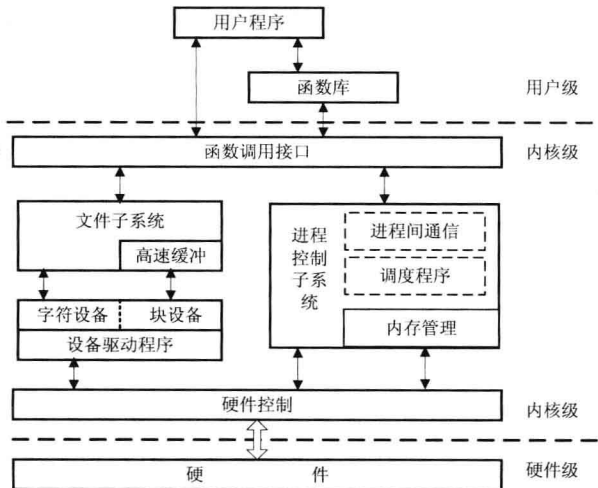


图 1-4 Linux 内核结构图

1.1.5 Linux 用户管理

Linux 系统是一个多用户系统，能做到不同的用户同时访问不同的文件。因此，一定要有文件权限控制机制。

Linux 中每一个用户有一个唯一的用户 ID。Linux 下的用户分为三类：超级用户、系统用户和普通用户。

超级用户的用户名为 root，它拥有一切权限。

系统用户是 Linux 系统正常工作所必需的内建用户，主要是为了满足相应的系统进程对文件属主的要求而建立的，系统用户不能用来登录，如 bin、daemon、adm、lp 等用户。

普通用户是为了让使用者能够使用 Linux 系统资源而建立的，大多数用户属于此类。

一个用户属于一个用户组，也可以属于多个用户组，Linux 中每一个组有唯一的组 ID。

在 Linux 操作系统中，超级用户 (root) 的权限是最高的，也被称为超级权限的拥有者。普通用户无法执行的操作，root 用户都能完成，所以也被称为超级管理用户。在系统中，每个文件、目录和进程都归属于某一个用户，没有用户许可，其他普通用户是无法操作的，但对 root 除外。

环境变量指该用户环境中定义生效的变量。用户环境变量定义在 .profile 和 .bash_profile 文件中，支持递归定义。用户可以在用户环境下用 env 命令查看环境变量。

1.1.6 Linux 文件管理

Linux 设计哲学继承了 UNIX 血统，一切皆为文件。Linux 文件包括如下类型：

- d: 目录文件，目录也是一种文件 (directory)。
- l: 符号链接 (指向另一个文件) (link)。
- b: 块设备文件 (block device)。
- c: 字符设备文件 (character device)。
- p: 命名管道文件 (named pipe)。
- s: 套接字文件 (socket)。
- -: 普通文件，或者更准确地说，不属于以上几种类型的文件。

Linux 中的文件属性如图 1-5 所示。文件属性通常有 10 位，第 1 位为该文件类型，第 2~4 位为该文件用户的权限，第 5~7 位为该组用户的权限，第 8~10 位是其他用户权限。

首先，Linux 中的文件按用户权限可分为三个不同的用户级别：文件拥有者 (u)、所属的用户组 (g) 和系统里的其他用户 (o)。每类用户级别对文件又可以定义三种不同的访问权限：可读 (r)、可写 (w) 和可执行 (x)。具体说明如下。