

# 嵌入式Linux实时操作系统及应用编程

熊茂华 熊 昕 编著  
周顺先 主审

清华大学出版社

# 嵌入式Linux实时操作系统及应用编程

熊茂华 熊 昕 编著  
周顺先 主审

清华大学出版社  
北 京

## 内 容 简 介

本书的内容包括：嵌入式 Linux 操作系统简介、Linux 常用命令集、嵌入式 Linux 的开发环境及工具软件、嵌入式 Linux 的构建、嵌入式 Linux 的 Shell 编程、嵌入式 Linux 的内存管理、嵌入式 Linux 下文件的操作、嵌入式 Linux 进程与进程调度、嵌入式 Linux 多线程开发、嵌入式 Linux 网络编程和 GTK+ 图形界面编程。本书是嵌入式 Linux 实时操作系统及应用编程的一本实用指导书籍，通过案例详细介绍嵌入式 Linux 实时操作系统的应用编程，案例中的程序都取自实际的项目，且对程序有详细注解。本书配套的案例程序、课件及每章的习题参考答案可从 <http://www.tup.com.cn> 下载。

本书深入浅出，可作为高等院校相关课程的本科生和高职生教材，也可作为研究生和嵌入式系统编程人员的技术参考书。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

### 图书在版编目(CIP)数据

嵌入式 Linux 实时操作系统及应用编程/熊茂华,熊昕编著. —北京:清华大学出版社, 2011.5

ISBN 978-7-302-25075-3

I. ①嵌… II. ①熊… ②熊… III. ①Linux 操作系统 IV. ①TP316.89

中国版本图书馆 CIP 数据核字(2011)第 040305 号

责任编辑：孟毅新

责任校对：袁芳

责任印制：王秀菊

出版发行：清华大学出版社

地 址：北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编：100084

社 总 机：010-62770175

邮 购：010-62786544

投稿与读者服务：010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈：010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者：清华大学印刷厂

经 销：全国新华书店

开 本：185×260 印 张：27 字 数：638 千字

版 次：2011 年 5 月第 1 版 印 次：2011 年 5 月第 1 次印刷

印 数：1~3000

定 价：48.00 元

产品编号：035304-01

在世界范围内,社会经济的发展产生了一些新的需求,这也促进了嵌入式技术的广泛应用。而中国正在成为世界制造大国,在消费电子、工业应用、军事国防、网络设备等领域都有嵌入式系统的应用,同时嵌入式技术反过来刺激了许多新的应用需求,如在信息家电、医疗电子病历、微小型智能武器等领域,嵌入式技术的应用日益广泛,相应的企业针对嵌入式技术人才的需求也越来越大。因此近几年来,各高职高专院校也开始增设了嵌入式技术应用专业。但是,各院校在嵌入式技术应用专业教学过程中面临教材难觅的困境。针对教材缺乏的情况,我们调研了几十所已开设了“嵌入式技术应用”专业的高职高专院校的嵌入式技术人才培养方案、课程设置、教材建设与开发、学生的学习情况及嵌入式技术应用人才就业现状,通过认真细致的整理、分析和研讨,编写了这套适用于高等院校嵌入式技术人才培养规划的系列教材,包括:《ARM9 嵌入式系统设计与开发应用》、《ARM 体系结构与程序设计》、《嵌入式 Linux C 语言应用程序设计与实践》、《嵌入式操作系统与编程》、《嵌入式 Linux 实时操作系统及应用编程》、《嵌入式  $\mu\text{C}/\text{OS}-\text{II}$  实时操作系统及应用编程》、《嵌入式 Windows CE 应用开发技术》、《嵌入式概论》、《嵌入式系统项目分析》共 9 本。

本书的主要内容如下。

第 1 章 对嵌入式 Linux 操作系统做一些简单介绍,对 Linux 的特征及组成、Linux 内核、常用的嵌入式操作系统、嵌入式系统开发过程等分别进行阐述。

第 2 章 介绍 Linux 常用命令集,主要包括 Linux 常用命令、系统管理命令、文件管理命令、文件传输与编辑命令、目录及其操作命令、文件压缩与磁盘挂载命令及网络操作命令等。

第 3 章 介绍嵌入式 Linux 的开发环境及工具软件,主要内容是嵌入式 Linux 的开发环境、Linux 开发工具的使用、GNU make 命令和 Makefile 文件、使用 Autoconf 和 Automake 生成 Makefile 文件、嵌入式 Linux 编辑器 vi 的使用、Emacs 综合编辑器等。

第 4 章 介绍嵌入式 Linux 的构建,主要内容包括嵌入式 BootLoader 技术、嵌入式 Linux 根文件系统构建、Linux 内核及启动过程、基于 ARM 目标板的 Linux 系统及应用程序的烧写等知识。

第 5 章 介绍嵌入式 Linux 的 Shell 编程,主要包括 Shell 编程基础、Bash 的内部命令、Shell 程序的运行与调试等知识。

第 6 章 介绍嵌入式 Linux 的内存管理,主要包括 Linux 内存管理的基本概念及相

关数据结构、Linux 的进程与内存管理、虚存空间的管理、共享内存、动态内存分配及 malloc/free 的实现、Linux 内存的使用等知识。

第 7 章 介绍嵌入式 Linux 下文件的操作,主要包括虚拟文件系统(VFS)、通用文件模型、不带缓存的文件 I/O 操作、Linux 标准 I/O 流、文件锁的应用等知识。

第 8 章 介绍嵌入式 Linux 进程与进程调度,主要包括嵌入式 Linux 进程描述符及任务结构、进程的调度、嵌入式 Linux 进程控制相关 API、嵌入式 Linux 进程间通信 API、Linux 中进程间通信、管道、信号通信、进程间的通信——共享内存、消息队列、Linux 守护进程等知识。

第 9 章 介绍嵌入式 Linux 多线程开发,主要包括 Linux 多线程相关 API、信号灯、互斥量、条件变量等知识。

第 10 章 介绍嵌入式 Linux 网络编程,主要包括 TCP/IP 的分层模型、网络编程基础、Socket 网络编程、面向连接的客户/服务器模型及编程、无连接的客户/服务器模型及编程等知识。

第 11 章 介绍 GTK+ 图形界面编程,主要包括 Linux 下的图形界面编程、GTK+ 图形界面程序、消息和回调函数、GTK+ 的面向对象机制等知识;同时介绍排列控件、进度条、微调按钮、组合框和分栏列表、对话框控件、菜单控件等应用知识。

本书由熊茂华、熊昕共同编著,熊茂华负责全面内容规划、编排,周顺先副教授主审。熊茂华主要编写第 1 章、第 6~11 章,熊昕主要编写第 2~5 章。在本书的编写过程中,部分内容基于北京博创公司 UP-NETARM2410 嵌入式系统教学系统,在此谨向他们深表谢意。

由于时间紧迫,疏漏之处在所难免,欢迎各位老师和同学指正。

编 者

2011 年 5 月

<b>第 1 章 嵌入式 Linux 操作系统简介</b> .....	1
1.1 Linux 的特征及组成 .....	1
1.1.1 Linux 操作系统及其历史 .....	1
1.1.2 Linux 系统的特点和组成 .....	2
1.1.3 Linux 的内核版本与发行版本 .....	4
1.2 Linux 的内核分析 .....	4
1.2.1 Linux 内核的结构 .....	4
1.2.2 Linux 内核的作用 .....	5
1.2.3 Linux 内核的抽象结构 .....	5
1.2.4 Linux 内核的源代码 .....	7
1.3 常用的嵌入式操作系统 .....	10
1.3.1 嵌入式操作系统的概念和分类 .....	10
1.3.2 Linux .....	10
1.3.3 $\mu\text{C}/\text{OS-II}$ .....	11
1.3.4 Windows CE .....	11
1.4 嵌入式系统开发过程 .....	12
1.4.1 嵌入式系统的总体结构 .....	12
1.4.2 嵌入式系统的开发特点 .....	13
1.4.3 嵌入式系统的开发流程 .....	13
1.4.4 调试嵌入式系统 .....	14
习题 .....	17
<b>第 2 章 Linux 常用命令集</b> .....	18
2.1 系统管理命令 .....	18
2.2 文件管理命令 .....	26
2.3 文件传输与编辑命令 .....	31
2.4 目录及其操作命令 .....	32
2.5 文件压缩与磁盘挂载命令 .....	35



2.6 网络操作命令	38
习题	45
<b>第3章 嵌入式 Linux 的开发环境及工具软件</b>	<b>48</b>
3.1 嵌入式 Linux 的开发环境	48
3.1.1 嵌入式 Linux 开发环境建立	48
3.1.2 嵌入式 Linux 开发的一般过程	49
3.2 Linux 开发工具的使用	50
3.2.1 Linux 开发工具 GNU GCC 的使用	50
3.2.2 GDB 调试器简介	58
3.3 GNU make 命令和 Makefile 文件	62
3.3.1 Makefile 文件的规则	63
3.3.2 Makefile 文件的隐含规则	65
3.3.3 Makefile 文件的命令	68
3.3.4 Makefile 文件的变量	69
3.3.5 Makefile 文件的条件判断	72
3.3.6 Makefile 文件的常用函数	74
3.3.7 子目录 Makefile	76
3.4 使用 Automake 和 Autoconf 生成 Makefile 文件	77
3.4.1 工具软件	78
3.4.2 configure 的创建	78
3.4.3 编辑 configure.in 脚本	79
3.4.4 Automake 和 Autoconf 生成 Makefile	81
3.4.5 GNU M4 的宏	85
3.5 嵌入式 Linux 编辑器 vi 的使用	87
3.5.1 vi 的基本模式	87
3.5.2 vi 的基本操作	88
3.6 Emacs 综合编辑器	88
3.6.1 Emacs 的启动与退出	89
3.6.2 Emacs 的基本编辑	89
3.6.3 Emacs 的 C 模式	91
3.6.4 Emacs 的 Shell 模式	91
习题	92
<b>第4章 嵌入式 Linux 的构建</b>	<b>95</b>
4.1 嵌入式 BootLoader 技术	95
4.1.1 BootLoader 的概念	95
4.1.2 BootLoader 引导程序	97

4.1.3	vivi 简介 .....	100
4.1.4	vivi 代码分析 .....	103
4.2	嵌入式 Linux 根文件系统构建 .....	106
4.2.1	Linux 文件系统介绍 .....	106
4.2.2	根文件系统的目录结构 .....	107
4.2.3	使用 BusyBox 生成工具集 .....	109
4.2.4	根文件系统启动流程 .....	111
4.3	Linux 内核及启动过程 .....	112
4.3.1	Linux 内核结构 .....	113
4.3.2	Linux 内核配置、编译 .....	114
4.3.3	基于 ARM 开发板的 Linux 内核移植 .....	118
4.4	基于 ARM 目标板的 Linux 系统及应用程序的烧写 .....	122
4.4.1	JTAG 烧写 vivi .....	122
4.4.2	串口下载烧写 .....	123
4.4.3	应用程序下载到 ARM 目标板 .....	126
	习题 .....	127
<b>第 5 章</b>	<b>嵌入式 Linux 的 Shell 编程 .....</b>	<b>130</b>
5.1	Shell 简介 .....	130
5.2	Shell 编程基础 .....	132
5.2.1	环境变量与 Shell 变量 .....	132
5.2.2	Shell 特殊字符 .....	135
5.2.3	Shell 程序设计的流程控制 .....	137
5.3	Bash 的内部命令 .....	144
5.4	Shell 程序的运行与调试方法 .....	145
5.5	Shell 编程实例 .....	148
	习题 .....	151
<b>第 6 章</b>	<b>嵌入式 Linux 的内存管理 .....</b>	<b>153</b>
6.1	Linux 内存管理的基本概念及相关数据结构 .....	153
6.1.1	嵌入式操作系统内存管理机制 .....	153
6.1.2	Linux 内存管理的基础知识 .....	156
6.1.3	Linux 内存管理相关数据结构 .....	157
6.2	Linux 的进程与内存管理 .....	159
6.2.1	进程内存管理 .....	159
6.2.2	系统物理内存管理 .....	161
6.3	虚存空间的管理 .....	166
6.3.1	内核空间和用户空间 .....	166



6.3.2	进程的虚存区域	167
6.4	共享内存	168
6.4.1	共享内存的数据结构	168
6.4.2	共享内存的操作	170
6.5	动态内存分配及 malloc/free 的实现	174
6.6	Linux 内存的使用	176
	习题	184
<b>第 7 章</b>	<b>嵌入式 Linux 下文件的操作</b>	<b>185</b>
7.1	Linux 文件 I/O 系统概述	185
7.1.1	虚拟文件系统	185
7.1.2	通用文件模型	186
7.2	不带缓存的文件 I/O 操作	196
7.2.1	文件的创建和读写	196
7.2.2	移动文件的读写位置	201
7.2.3	文件的各个属性	201
7.2.4	目录文件的操作	202
7.3	Linux 标准 I/O 流	204
7.4	嵌入式 Linux 的文件操作应用举例	216
7.4.1	文件锁的应用	216
7.4.2	嵌入式 Linux 串行口通信设计	222
	习题	232
<b>第 8 章</b>	<b>嵌入式 Linux 进程与进程调度</b>	<b>234</b>
8.1	嵌入式 Linux 进程管理	234
8.1.1	进程描述符及任务结构	234
8.1.2	进程的调度	239
8.2	嵌入式 Linux 进程控制相关 API	242
8.3	嵌入式 Linux 进程间通信 API	252
8.3.1	Linux 中进程间通信	252
8.3.2	管道	254
8.3.3	命名管道	257
8.3.4	信号通信	262
8.3.5	进程间的通信——共享内存	277
8.3.6	消息队列	279
8.4	Linux 守护进程	286
8.4.1	守护进程概述	286
8.4.2	编写规则	286

8.4.3 守护进程实例.....	288
习题.....	289
<b>第9章 嵌入式Linux多线程开发.....</b>	<b>292</b>
9.1 Linux多线程相关API.....	293
9.1.1 线程的创建.....	293
9.1.2 线程的终止.....	295
9.1.3 线程的标识.....	298
9.1.4 线程的一次性初始化.....	298
9.1.5 线程的私有数据.....	299
9.2 信号灯.....	301
9.2.1 POSIX有名信号灯的API函数.....	301
9.2.2 POSIX基于内存的信号灯的API函数.....	306
9.3 互斥量.....	308
9.4 条件变量.....	316
习题.....	321
<b>第10章 嵌入式Linux网络编程.....</b>	<b>322</b>
10.1 TCP/IP协议简介.....	322
10.1.1 TCP/IP的分层模型.....	322
10.1.2 TCP/IP分层模型特点.....	323
10.1.3 Internet层中的协议.....	324
10.1.4 传输层的协议.....	327
10.2 网络编程基础.....	330
10.2.1 Socket概述.....	330
10.2.2 Socket基础.....	331
10.3 Socket网络编程.....	336
10.3.1 Socket基本函数.....	336
10.3.2 面向连接的客户/服务器模型及编程.....	342
10.3.3 无连接的客户/服务器模型及编程.....	345
习题.....	359
<b>第11章 GTK+图形界面编程.....</b>	<b>361</b>
11.1 Linux下的图形界面编程.....	361
11.1.1 GTK+开发环境.....	361
11.1.2 GTK+简介.....	365
11.2 GTK+图形界面程序的简单应用实例.....	366
11.3 消息和回调函数.....	371

11.4	GTK+的面向对象机制 .....	373
11.5	排列控件 .....	377
11.5.1	使用 box 排列控件 .....	377
11.5.2	使用 table 排列控件 .....	383
11.5.3	使用 arrows 控件 .....	389
11.5.4	工具提示对象 .....	391
11.6	进度条、微调按钮、组合框和分栏列表 .....	391
11.6.1	进度条 .....	391
11.6.2	微调按钮 .....	398
11.6.3	组合框 .....	399
11.6.4	分栏列表 .....	402
11.7	对话框控件 .....	411
11.8	菜单控件 .....	414
	习题 .....	418
	<b>参考文献</b> .....	<b>420</b>

本章将介绍嵌入式 Linux 操作系统基本知识。学习完本章,读者将了解以下内容。

- (1) Linux 的特征及组成。
- (2) Linux 内核的主要组成部分。
- (3) Linux 内核分析。
- (4) 常用的嵌入式 Linux 操作系统。

## 1.1 Linux 的特征及组成

### 1.1.1 Linux 操作系统及其历史

Linux 是一个功能强大的操作系统,同时它是一个自由软件,是免费的、开放源代码的,编制它的目的是建立不受任何商品化软件版权制约的,全世界都能自由使用的 UNIX 兼容产品。各种使用 Linux 作为内核的 GNU 操作系统正被广泛地使用着,虽然这些系统通常被称为 Linux,但是它们应该更精确地称为 GNU/Linux 系统。

Linux 最初是由芬兰赫尔辛基大学计算机系大学生 Linus Torvalds,在从 1990 年年底到 1991 年的几个月中,为了他自己的操作系统课程和后来的上网用途而编写的,在他自己买的 Intel 386 PC 上,利用 Tanenbaum 教授自行设计的微型 UNIX 操作系统 Minix 作为开发平台。Linus 说,刚开始的时候他根本没有想到要编写一个操作系统的内核,更是绝对没有想到这一举动会在计算机界产生如此重大的影响。最开始是一个进程切换器,然后是为他自己上网需要而自行编写的终端仿真程序,再后来是为他从网上下载文件的需要而自行编写的硬盘驱动程序和文件系统,这时他发现他已经实现了一个几乎完整的操作系统内核。

出于对这个内核的信心和美好的奉献精神与发展希望,Linus 希望这个内核能够免费使用,但出于谨慎他并没有在 Minix 新闻组中公布它,而只是于 1991 年年底在赫尔辛基技术大学的一台 FTP 服务器上发了一则消息,说用户可以下载 Linux 的公开版本(基于 Intel 386 体系结构)和源代码。从此以后,奇迹开始发生了。

Linux 的兴起可以说是 Internet 创造的一个奇迹。到 1992 年 1 月止,全世界只有

100 个左右的人在使用 Linux,但由于它是在 Internet 上发布的,网上的任何人在任何地方都可以得到 Linux 的基本文件,并可通过电子邮件发表评论或者提供修正代码,这些 Linux 的热心者中有将之作为学习和研究对象的大专院校的学生和科研机构的科研人员,也有网络黑客等,他们所提供的全部初期上载的代码和评论,后来证明对 Linux 的发展至关重要。正是在这众多热心者的努力下,Linux 在不到 3 年的时间里成为一个功能完善、稳定可靠的操作系统。

在 Linux 诞生不久之后,Linus Torvalds 将其贡献给了自由软件社区,Linus 本人认为“使 Linux 成为 GPL 的一员是我一生中做过最漂亮的一件事”。

Linux 操作系统可以说是 UNIX 操作系统的克隆体,自 Linus Torvalds 创建 Linux 开始,就允许其他人免费地自由运用该系统的源代码,并且鼓励其他人进一步对其进行开发。Linux 操作系统继承了 UNIX 操作系统 25 年的经验、源代码以及技术支持,它在短短的几年内,得到了非常迅猛的发展,很快成为最受人们喜爱的操作系统之一,而这一点正是与它本身具有的良好特性分不开的。

## 1.1.2 Linux 系统的特点和组成

### 1. Linux 系统的特点

Linux 操作系统在短短的几年之内得到了非常迅猛的发展,这与 Linux 具有的良好特性是分不开的。Linux 包含了 UNIX 的全部功能和特性。简单地说,Linux 具有以下主要特点。

(1) 开放性。指系统遵循世界标准规范,特别是遵循开放系统互联(Open System Interconnect,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,可方便地实现互联。另外,源代码开放和免费,使得获取 Linux 非常方便,而且使用 Linux 可节省费用。Linux 开放源代码,使用者能控制源代码,按照需要对部件混合搭配,建立自定义扩展。

(2) 多用户。指系统资源可以被不同用户各自拥有使用,即每个用户对自己的资源(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最主要的一个特点,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 系统调度每一个进程平等地访问微处理器。

(4) 出色的速度性能。Linux 可以连续运行数月、数年而无须重新启动,与 Windows NT(经常死机)相比,这一特点尤其突出。即使作为一种台式机操作系统,与许多用户非常熟悉的 UNIX 相比,它的性能也显得更为优秀。Linux 不太在意 CPU 的速度,它可以把处理器的性能发挥到极限(用户会发现,影响系统性能提高的限制因素主要是其总线和磁盘 I/O 的性能)。

(5) 良好的用户界面。Linux 向用户提供了 3 种界面:用户命令界面、系统调用界面和图形用户界面。

(6) 丰富的网络功能。Linux 是在 Internet 基础上产生并发展起来的,因此,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。

(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时它仍然能按其自身方式运行的能力。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和任何平台上运行。可移植性为运行 Linux 的不同计算机平台与其他任何计算机进行准确而有效的通信提供了手段,不需要另外增加特殊和昂贵的通信接口。

(9) 具有标准兼容性。Linux 是一个与 POSIX(Portable Operating System Interface)相兼容的操作系统,它所构成的子系统支持所有相关的美国国家标准学会(American National Standards Institute,ANSI)、ISO、互联网工程任务组(Internet Engineering Task Force,IETF)和万维网联盟(World Wide Web Consortium,W3C)业界标准。为了使 UNIX System V 和伯克利软件套件(Berkeley Software Distribution,BSD)上的程序能直接在 Linux 上运行,Linux 还增加了部分 System V 和 BSD 的系统接口,使 Linux 成为一个完善的 UNIX 程序开发系统。Linux 也符合 X/Open 标准,具有完全自由的 X Window 功能。另外,Linux 在对工业标准的支持上做得非常好,由于各个 Linux 发布厂商都能自由获取和接触 Linux 的源代码,各厂商发布的 Linux 仍然缺乏标准,不过这些差异非常小。它们的差异主要存在于所捆绑应用程序的版本、安装工具的版本和各种系统文件所处的目录结构。

## 2. Linux 系统的组成

Linux 一般有 4 个主要部分:内核、Shell、应用程序和文件系统。内核、Shell 和文件系统一起形成了基本的操作系统结构。它们使得用户可以运行程序、管理文件并使用系统。

(1) Linux 内核。内核(Kernel)是系统的核心,实现操作系统的基本功能。内核需要两个方面的支持,即硬件方面和软件方面。

① 硬件方面。控制硬件设备,管理内存,提供硬件接口,处理基本的输入/输出(I/O)。

② 软件方面。管理文件系统,为程序分配内存和 CPU 时间等。

(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。Shell 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执行。Shell 编程语言具有普通编程语言的很多特点,用这种编程语言编写 Shell 程序与其他应用程序具有同样的效果。目前常见的 Shell 有 Bourne Shell(sh)、Korn Shell(ksh)、C Shell(csh)、Bourne-again Shell(bash)。

(3) Linux 应用程序。标准的 Linux 系统都有一套称为应用程序的程序集,包括文本编辑器、编程语言、X Window、办公套件、Internet 工具、数据库等。当然,还可以有用户自己编写的具有特定功能的应用程序。

(4) Linux 文件系统。文件系统是文件存放在磁盘等存储设备上的组织方法,通常按照目录层次的方式进行组织。每个目录可包括多个子目录以及文件,系统以“/”为根目录。系统中的所有数据都存储在文件系统中以使用户读取、查询和写入。Linux 能支持多种目前流行的文件系统,如 Ext2、Ext3、FAT、VFAT、ISO 9660、NFS 等。

### 1.1.3 Linux 的内核版本与发行版本

Linux 有内核版本和发行版本之分。

#### 1. 内核版本

内核版本是在 Linus 领导下的开发小组开发出的系统内核的版本号。内核版本号由 3 个数字组成: r. x. y。其中, r: 目前发布的 Kernel 主版本; x: 一般来说, x 位为偶数的版本表明这是一个可以使用的稳定版本, 如 2. 4. 4, x 位为奇数的版本一般加入了一些新的内容, 不一定很稳定, 是测试版本, 如 2. 1. 111; y: 错误修补的次数, 如 CentOS 5 使用的内核版本是 2. 6. 18。

**注意:** 时至今日, Linux 的内核仍旧由 Linus Torvalds 领导下的开发小组维护。可以访问 <http://www.kernel.org/> 获得最新的内核信息。

#### 2. 发行版本

发行版本是一些组织或厂家将 Linux 系统内核与应用软件和文档封装起来, 并提供一些安装界面和系统设定管理工具的一个软件包的集合。目前已经有了几百种发行版本。相对于内核版本, 发行版本的版本号随着发布者的不同而不同, 与系统内核的版本号是相对独立的。常见的发行版本有以下 3 类。

##### (1) 社区发布版本

CentOS: <http://www.centos.org/>

Ubuntu: <http://www.ubuntu.com/>

Debian: <http://www.debian.org/>

OpenSUSE: <http://www.opensuse.org/>

Fedora: <http://www.fedoraproject.org/>

Gentoo: <http://www.gentoo.org/>

##### (2) CD Live 版本

Ubuntu: <http://www.ubuntu.com/>

Finnix: <http://www.finnix.org/>

Knoppix: <http://www.knoppix.org/>

##### (3) 商业支持版本

Red Hat Enterprise: <http://www.redhat.com/>

SUSE: <http://www.novell.com/products/>

Mandriva: <http://www.mandrivalinux.com/>

Turbolinux: <http://www.turbolinux.com/>

## 1.2 Linux 的内核分析

### 1.2.1 Linux 内核的结构

Linux 的内核不是孤立的, 必须把它放在整个系统中去研究, 图 1.1 显示了 Linux 内

核在整个操作系统的位置。

由图 1.1 可以看出, Linux 操作系统由以下 4 个部分组成。

(1) 用户进程。用户应用程序是运行在 Linux 操作系统最高层的一个庞大的软件集合, 当一个用户程序在操作系统之上运行时, 它成为操作系统中的一个进程。

(2) 系统调用接口。在应用程序中, 可通过系统调用来调用操作系统内核中特定的过程, 以实现特定的服务。例如, 在程序中安排一条创建进程的系统调用, 则操作系统内核便会为之创建一个新进程。

系统调用本身也是由若干条指令构成的过程。但它与一般的过程不同, 主要区别是: 系统调用运行在内核态(系统态), 而一般过程运行在用户态。在 Linux 中, 系统调用是内核代码的一部分。

(3) Linux 内核。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。简而言之, 内核实际是抽象的资源操作到具体硬件操作细节之间的接口。

(4) 硬件。这个子系统包括了 Linux 安装时需要的所有可能的物理设备。例如, CPU、内存、硬盘、网络硬件等。

上面的这种划分把整个 Linux 操作系统分为 4 个层次。把用户进程也纳入操作系统的范围内是因为用户进程的运行和操作系统密切相关, 而系统调用接口可以说是操作系统内核的扩充, 硬件则是操作系统内核赖以生存的物质条件。这 4 个层次的依赖关系表现为上层依赖下层。

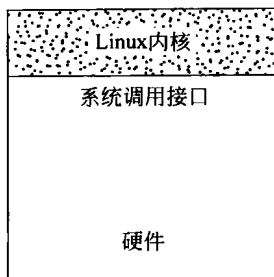


图 1.1 Linux 内核在整个操作系统中的位置

## 1.2.2 Linux 内核的作用

从程序员的角度来讲, 操作系统的内核提供了一个与计算机硬件等价的扩展或虚拟的计算平台。它抽象了许多硬件细节, 程序可以以某种统一的方式进行数据处理, 而程序员则可以避开许多硬件细节。从另一个角度讲, 普通用户则把操作系统看成是一个资源管理者, 在它的帮助下, 用户可以以某种易于理解的方式组织自己的数据, 完成自己的工作并和其他人共享资源。

Linux 以统一的方式支持多任务, 而这种方式对用户进程是透明的, 每一个进程运行起来就好像只有它一个进程在计算机上运行一样, 独占内存和其他的硬件资源, 而实际上, 内核在并发地运行几个进程, 并且能够让几个进程公平合理地使用硬件资源, 也能使各进程之间互不干扰安全地运行。

## 1.2.3 Linux 内核的抽象结构

Linux 内核由 5 个主要的子系统组成, 如图 1.2 所示。

(1) 进程调度控制着进程对 CPU 的访问。当需要选择下一个进程运行时, 由调度程序选择最值得运行的进程。可运行进程实际是仅等待 CPU 资源的进程, 如果某个进程在等待其他资源, 则该进程是不可运行进程。Linux 使用了比较简单的基于优先级的进



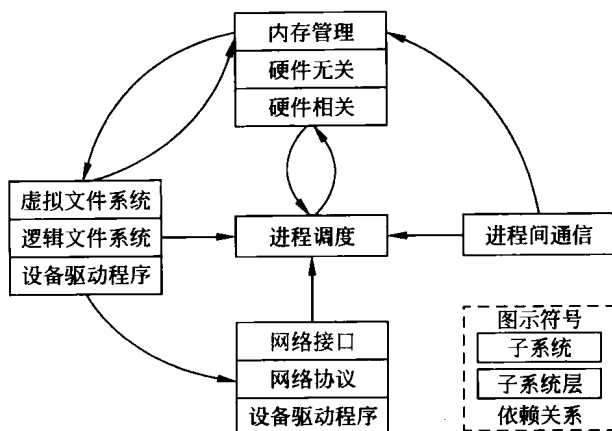


图 1.2 Linux 内核子系统及其之间的关系

程调度算法选择新的进程。

(2) 内存管理(Memory Management, MM)允许多个进程安全地共享主内存区域。Linux 的内存管理支持虚拟内存,即在计算机中运行的程序,其代码、数据和堆栈的总量可以超过实际内存的大小,操作系统只将当前使用的程序块保留在内存中,其余的程序块则保留在磁盘上。必要时,操作系统负责在磁盘和内存之间交换程序块。

内存管理从逻辑上可以分为硬件无关的部分和硬件相关的部分。硬件无关的部分提供了进程的映射和虚拟内存的对换;硬件相关的部分为内存管理硬件提供了虚拟接口。

(3) 虚拟文件系统(Virtual File System, VFS)隐藏了各种不同硬件的具体细节,为所有设备提供了统一的接口,VFS 还支持数十种不同的文件系统,这也是 Linux 较有特色的一部分。

虚拟文件系统可分为逻辑文件系统和设备驱动程序。逻辑文件系统指 Linux 所支持的文件系统,如 Ext2、FAT 等;设备驱动程序指为每一种硬件控制器所编写的设备驱动程序模块。

(4) 网络接口提供了对各种网络标准协议的存取和各种网络硬件的支持。网络接口可分为网络协议和网络设备驱动程序两部分。网络协议部分负责实现每一种可能的网络传输协议,网络设备驱动程序负责与硬件设备进行通信,每一种可能的硬件设备都有相应的设备驱动程序。

(5) 进程间通信(Inter-Process Communication, IPC)支持进程间各种通信机制。

从图 1.2 可以看出,处于中心位置的是进程调度,所有其他的子系统都依赖于它,因为每个子系统都需要挂起或恢复进程。一般情况下,当一个进程等待硬件操作完成时,它被挂起;当操作真正完成时,进程被恢复执行。例如,当一个进程通过网络发送一条消息时,网络接口需要挂起发送进程,直到硬件成功地完成消息的发送,当消息被发送出去以后,网络接口给进程返回一个代码,表示操作的成功或失败。其他子系统(内存管理、虚拟文件系统及进程间通信)以相似的理由依赖于进程调度。各个子系统之间的依赖关系如下。