

北京科海培训中心



CD-ROM

嵌入式 Linux 系统设计与应用

- ◆ 如何开发具体的嵌入式 Linux 系统
- ◆ 如何在嵌入式 Linux 系统上开发应用程序

6

王学龙 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

北京科海培训中心

嵌入式 Linux 系统设计与应用

王学龙 编著

- 如何开发具体的嵌入式 Linux 系统
- 如何在嵌入式 Linux 系统上开发应用程序

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书分 4 部分详细介绍嵌入式 Linux 系统的设计及应用实例。第 1 部分集中对 Linux 系统的内核源代码进行详细分析,重点放在与嵌入式系统有关的若干问题上。第 2 部分介绍如何在嵌入式 Linux 系统中编程,包括与 VOIP 应用有关的 GNU C++ 编程、多线程编程以及 BASH 和 GCC 编程。第 3 部分对如何修改源代码,如何组建嵌入式 Linux 系统进行实例分析,同时对控制和修改与系统有关的脚本进行示范。最后设计了嵌入式 Linux 系统的文件系统。第 4 部分逐步介绍嵌入式 Linux 系统的一个应用实例,即如何在嵌入式 Linux 系统中开发 VOIP(Voice on IP)应用程序,并且给出此应用程序范例的源代码。

开发嵌入式 Linux 系统的新手以及 Linux 系统的高级用户均可以本书作为参考书。

版权所有,盗版必究。

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。

书 名: 嵌入式 Linux 系统设计与应用

作 者: 王学龙

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)

印刷者: 北京门头沟胶印厂

发行者: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张: 26 字数: 632 千字

版 次: 2001 年 8 月第 1 版 2001 年 8 月第 1 次印刷

印 数: 0001~5000

盘 号: ISBN 7-900635-99-8

定 价: 45.00 元(光盘)

前 言

随着嵌入式系统应用的流行，各大企业已经纷纷将嵌入式系统的开发作为自己的研究发展战略，例如联想、华为和中兴公司等都成立了自己的嵌入式部门。而嵌入式 Linux 系统又是嵌入式系统中的宠儿，毕竟 Linux 系统是一个自由软件。作为嵌入式 Linux 系统应用研发大潮中的一员，作者深感在黑暗中摸索是多么困难，而且市场上目前还没有一本对嵌入式 Linux 系统进行全面分析的书籍，这一方面是由于嵌入式 Linux 系统技术性太强，只适合高级研发人员，一方面是由于嵌入式 Linux 系统的应用往往牵涉到商业机密不便向外透露。作者想在这种情况下献出自己的一部分经验抛砖引玉，希望本书能激起读者和用户对嵌入式 Linux 系统开发的兴趣，把握嵌入式系统应用的契机，为我国的科技发展做出更大贡献。

本书主要内容

本书将分 4 部分详细介绍嵌入式 Linux 系统的设计及应用实例。初级读者可以通过本书逐步接触到嵌入式 Linux 系统的点点滴滴，而资深的嵌入式系统开发人员也可以通过本书理清思路。尤其是本书介绍的应用实例，它是嵌入式系统与通信系统成功应用结合的范例。

第 1 部分“Linux 内核源代码分析”

本部分集中对 Linux 系统的内核源代码进行详细分析，由于本书致力于介绍嵌入式 Linux 系统的应用，因此这一部分将内核分析的重点放在与嵌入式系统有关的若干问题上。

- 第 1 章介绍 Linux 内核的进程管理，这里不仅有详尽的描述，还有核心代码文件的分析。理解嵌入式系统的进程管理，对于在该系统中创建进程项、进行多线程编程是十分有利的。实际上，不同的嵌入式系统，它们的进程管理机制大相径庭，例如：它们允许同时创建的进程个数就不一样，Linux 系统为 512 个。
- 第 2 章详细分析 Linux 内核的内存管理，即它的虚存管理机制。不同的操作系统管理内存的方法是不同的；即使同一操作系统，当它运行在不同的平台时它管理内存的方法也不一样。例如：Linux 系统在 X86 平台上使用二级分页机制，而在有些平台上使用三级分页机制。此外，为了满足实时性的要求，现在有些商家已经开发出了无虚拟内存的嵌入式 Linux 系统。
- 第 3 章和第 4 章对 Linux 内核的文件系统进行介绍。对于操作系统来说，文件系统也是它的一大特色之一。Windows 操作系统不同 Linux 操作系统，也体现在文件系统的不一致上。当然，可以肯定，不同的嵌入式系统，它们的文件管理方法也不同。嵌入式 Linux 系统则使用 VFS 文件系统接口和 EXT2 文件系统。实际上，用户也

可以在现有的 EXT2 文件系统上进行修改，在自己的嵌入式 Linux 系统中使用自定义的文件系统。

- 第 5 章简单介绍 Linux 内核的网络设备接口。与 Unix 系统内核一样，Linux 内核的网络通信是基于套接字 (socket) 进行的，它遵循 TCP/IP 协议。为什么把网络设备接口单列一章进行介绍呢？因为随着网络技术的发展，许多实际应用或多或少都与网络有一定联系，基于嵌入式 Linux 系统的应用也是如此。不仅这样，许多商业机构还制定了基于 TCP/IP 协议的应用协议。本书在第 3 部分介绍的应用实例就使用了基于 TCP/IP 协议的 H.323 协议族。
- 第 6 章讨论 Linux 内核的系统调用，有过 Linux 内核分析经验的用户已经知道，Linux 内核的系统调用不仅已用在内核的源代码中，而且它可以用于自己开发的应用程序中。正因为如此，对于开发嵌入式 Linux 系统的用户，应该理解 Linux 内核的系统调用原理，同时掌握 Linux 内核的系统调用接口。这对于在嵌入式 Linux 系统中开发应用是十分必要的。

第 2 部分“Linux 系统编程”

本部分对如何在嵌入式 Linux 系统中编程进行专题介绍。编程的重点则放在与第 4 部分所介绍的 VOIP 应用有关的 C++ 编程、多线程编程等上。这部分相当于开发嵌入式 Linux 系统的软件基础。

- 第 7 章是有关 Linux 系统中的 Bash 脚本编程的专题。在应用嵌入式 Linux 系统时，经常需要改动系统的运行脚本或者编写自己的脚本程序。例如：在进行 Gcc 或 GNU C++ 开发时需要开发自定义的 Makefile，而 Makefile 中有大量的 Bash 语法。
- 第 8 章是简单介绍 Gcc 编程的专题。Gcc 编程与标准 C 编程的区别不是很大，但是 Linux 系统下的 Gcc 有一些特别的编程、调试规则。使用 Gcc 编译多个源文件时可以使用 Makefile 文件。嵌入式 Linux 系统中的许多应用程序都是利用 Gcc 开发的。
- 第 9 章是讨论 GNU C++ 编程的专题。相对而言，GNU C++ 比 Gcc 更胜一筹，因为在 Linux 系统中，g++ 编译器不仅可以编译 GNU C++ 代码，而且可以编译 Gcc 代码。因此当利用 GNU C++ 编程时可以综合 GNU C++ 和 Gcc 优点。本书第 4 部分介绍的应用就是利用 GNU C++ 开发的。
- 第 10 章是介绍多线程编程的专题。支持多线程是多任务操作系统的特点。对于要求有较高实时性的场合，使用多线程是非常方便的。熟悉 PSOS 系统的用户对多线程编程一定有很深的理解，因为在 PSOS 系统中为多线程编程专门提供了调用库。但是 Linux 系统对多线程的处理与 PSOS 系统有很大区别。

第 3 部分“建立嵌入式 Linux 系统”

本部分对如何修改源代码，如何组建嵌入式 Linux 系统进行实例分析，同时对控制和修改与系统有关的脚本进行示范。最后设计了嵌入式 Linux 系统的文件系统。

- 第 11 章介绍如何编译 Linux 内核。本章首先讨论怎样改动 Linux 内核以符合自己的要求，然后奉献出自己的一点经验。最后对版本的选取、编译内核的步骤等进行

系统介绍。

- 第 12 章介绍 Linux 系统中的脚本控制。Linux 系统在启动和关闭时需要大量脚本，适当的控制、修改这些脚本不仅可以使文件系统简洁明了，而且可以大量节约内存。对于内存在 4M 左右的嵌入式 Linux 系统，如果不进行脚本控制，系统将无法完全启动。
- 第 13 章讨论自定义文件系统的组建。本章讨论的文件系统是 EXT2 文件系统，它不仅是 Linux 系统默认的文件系统，而且是嵌入式 Linux 系统中使用最广的文件系统。由于系统启动所需的脚本以及系统的命令文件均在文件系统内，所以具有文件系统是 Linux 系统启动的基础。

第 4 部分“嵌入式 Linux 系统的应用”

本部分逐步介绍嵌入式 Linux 系统的一个应用实例，即如何在嵌入式 Linux 系统中开发 VOIP(Voice on IP)应用。此部分将对 VOIP 应用的开发、调试给出详细的说明，并且给出此应用范例的源代码。

- 第 14 章介绍 IP 电话的实现原理。IP 电话属于语音通信的范畴，其应用的业务名称是 VoIP（即 Voice on IP）。目前开发语音通信产品的商家大多遵循 ITU（国际电信联盟）的 H 系列协议，与语音通信最相关的是 H.323 协议族。
- 第 15 章详细分析系统引导程序 PPCBOOT。计算机用户一般对 BIOS 有一定了解，在嵌入式系统中，则没有 BIOS 的概念。但是其中的系统引导程序与 BIOS 有相似的功能。本章不仅介绍如何修改 PPCBOOT 的源代码以适应自己的硬件体系结构，还说明如何编译这些源代码以及利用 PPCBOOT 引导系统的详细过程。这是建立开发平台的基础。
- 第 16 章讨论建立开发平台。确定开发模型是系统开发的第一步，本章介绍的开发模型是目前嵌入式 Linux 系统开发商使用最广的模型。我们将重点讨论 MontaVista 的开发工具包 CDK1.2。
- 第 17 章探讨在协议栈的基础上开发应用软件。本章将综合本书的所有知识开发一个实实在在的应用程序。此应用程序相当于一个 IP 电话终端软件。在此读者可理解到开发嵌入式 Linux 系统应用的技巧。本章旨在为开发嵌入式 Linux 系统应用或者其他嵌入式系统应用的用户提供示范，以帮助开发人员加快开发进度、少走弯路。

本书在应用部分不仅对嵌入式 Linux 系统高层应用进行了剖析，而且对嵌入式 Linux 系统的底层进行探讨。虽然本书没有更多的篇幅介绍嵌入式 Linux 系统的硬件设计，但软件部分的介绍不可避免地要涉及到 CPU 的结构，因此此部分对 MPC823 的应用有一定的涉及。此芯片是嵌入式系统应用中的热门芯片。

本书配套光盘说明

本书配套光盘提供了两方面的内容：

- 本书的部分源代码。
- 与本书内容相关的技术文档。与本书内容相关的技术文档主要包含在目录“技术文档”下，这些资料可以用于相关的嵌入式系统开发。与本书内容相关的技术文档包含了 etherboot、PSOS、Sniff、Minix 和 Tiny Linux。

本书由潇湘工作室策划并组织编写，王学龙主笔，另外还有许多人员做了大量的工作。我们希望读者在学习完本书之后，能迅速踏入嵌入式 Linux 系统开发的大门，成为嵌入式系统开发中的重要一员。若有问题，欢迎通过电子邮件 michaelwang@0451.com 与作者联系，作者将尽力解答而且提供其他 Linux 爱好者的邮件地址，共同探讨 Linux 系统开发问题。

目 录

第 1 部分 Linux 内核源代码分析

第 1 章 Linux 进程管理代码分析	3
1.1 基本概念	3
1.2 进程的状态和标志	4
1.3 Linux 中进程间的状态转换	9
1.3.1 进程的创建	9
1.3.2 进程的调度	10
1.3.3 进程间通信	12
1.3.4 进程从运行态到停止态的转换	24
1.3.5 进程从停止态到运行态的转换	25
1.3.6 进程的终止	26
1.4 小结	27
第 2 章 内存管理分析	29
2.1 概述	29
2.1.1 Linux 内存管理的基本特点	29
2.1.2 Linux 内存管理的主要实现技术	29
2.1.3 虚拟内存模型	31
2.1.4 页面管理	31
2.2 Linux 内存管理数据结构	35
2.2.1 32 位虚拟地址	35
2.2.2 Linux 的多级页表结构	35
2.2.3 页表项的格式	36
2.2.4 动态地址映射	37
2.2.5 用户进程的虚拟内存结构	38
2.3 进程内存管理数据结构的建立、维护及其系统调用	39
2.3.1 进程载入、创建及内存管理数据结构和链接建立	39
2.3.2 缺页中断服务	42
2.4 主要函数分析	44
2.4.1 内存管理的主要文件	44
2.4.2 内存映射文件	48
2.5 小结	55

第 3 章 虚拟文件系统	56
3.1 基本概念	56
3.2 VFS 的数据结构	57
3.2.1 VFS 的超级块结构	57
3.2.2 VFS inode 及其相关数据结构	59
3.3 高速缓存的管理	60
3.3.1 缓冲区高速缓存	60
3.3.2 VFS inode 高速缓存	66
3.3.3 VFS 目录高速缓存	67
3.4 VFS inode 资源链	69
3.4.1 资源链的组织及主要操作	69
3.4.2 VFS inode 的主要操作	70
3.5 文件系统的管理	73
3.6 文件系统的系统调用	74
3.6.1 sys_open 系统调用	74
3.6.2 sys_close () 系统调用	75
3.7 小结	76
第 4 章 EXT2 文件系统	77
4.1 概述	77
4.2 EXT2 文件系统的数据结构	78
4.2.1 EXT2 文件系统的物理布局	78
4.2.2 EXT2 文件系统的超级块结构	78
4.2.3 EXT2 文件系统的 inode 结构	81
4.2.4 EXT2 文件系统的组描述符	83
4.3 文件管理	84
4.3.1 系统打开文件表	84
4.3.2 EXT2 的目录项	85
4.3.3 文件的访问	85
4.3.4 数据块的管理	86
4.4 EXT2 文件系统的系统调用	88
4.4.1 sys_mkdir 系统调用	88
4.4.2 sys_rmdir 系统调用	90
4.5 小结	91
第 5 章 Linux 的网络设备接口	93
5.1 网络设备概述	93
5.2 网络设备的数据结构	95
5.3 网络设备的初始化	99

5.3.1 模块初始化模式的分析	99
5.3.2 启动初始化模式的分析	105
5.4 网络设备的打开和关闭	111
5.5 数据包的传输和接收	113
5.5.1 Socket 缓冲区及相关操作	114
5.5.2 数据包的传输	115
5.5.3 数据包的接收	117
5.6 网络设备驱动程序的开发	120
5.6.1 网络设备驱动程序的开发步骤	120
5.6.2 设备驱动程序的开发实例	120
5.7 小结	126
第 6 章 系统调用的原理	127
6.1 系统调用概述	127
6.2 Linux 系统调用流程	127
6.2.1 Linux 系统调用的中断机制	127
6.2.2 相关的数据结构及函数	128
6.2.3 Linux 系统调用的流程	136
6.3 系统调用的实例分析	141
6.3.1 mlock 系统调用简介	141
6.3.2 mlock 所用到的主要数据结构和重要常量	142
6.3.3 mlock 系统调用代码分析	145
6.4 系统调用的创建	155
6.4.1 系统调用编程	155
6.4.2 系统调用的连接和使用	156
6.5 小结	157

第 2 部分 Linux 系统编程

第 7 章 Bash 编程	161
7.1 Bash 的编程步骤	161
7.2 Bash 中的变量	162
7.2.1 特殊变量	163
7.2.2 Bash 中的引号	164
7.3 Bash 中的数值运算	164
7.4 Bash 中的特殊语句	166
7.4.1 条件语句	166
7.4.2 循环语句	168

7.4.3 移位语句	170
7.5 在 Bash 中使用函数	170
7.6 正则表达式	171
7.6.1 句点的单字符匹配	171
7.6.2 行首和行尾的匹配	172
7.6.3 星号和反斜杠的使用	172
7.6.4 括号匹配和指定次数匹配	173
7.6.5 Bash 脚本中常用的正则表达式	174
7.7 Bash 编程实例	175
7.7.1 网络检测脚本	175
7.7.2 备份脚本	175
7.7.3 限制访问脚本	181
7.7.4 清除日志脚本	183
7.8 小结	184
第 8 章 GNU make 管理项目与 Gcc 编程	185
8.1 make 管理项目简介	185
8.2 编写 Makefile 的规则	186
8.2.1 虚拟目标	187
8.2.2 Makefile 的变量	187
8.2.3 make 的变量	188
8.2.4 隐式规则	189
8.2.5 模式规则	190
8.3 make 命令	190
8.4 Makefile 实例分析	191
8.5 Gcc 的基本概念	197
8.6 Gcc 命令	199
8.7 Gcc 扩展简介	202
8.8 Gcc 编程实例分析	203
8.8.1 串行口通信程序	203
8.8.2 网络通信程序	205
8.9 小结	207
第 9 章 GNU C++ 编程	208
9.1 Gcc 与 GNU C++ 的比较	208
9.2 GNU C++ 程序概述	209
9.3 GNU C++ 的编译和调试	211
9.3.1 警告与错误	211
9.3.2 编译选项	212

9.3.3 调试技术	217
9.4 GNU C++编程特点	221
9.4.1 内存管理的特点	221
9.4.2 指针使用的特点	222
9.4.3 异常处理的特点	223
9.4.4 类对象复制的特点	226
9.5 GNU C++程序实例分析	227
9.5.1 接收变元的程序示例	227
9.5.2 类对象复制程序示例	229
9.5.3 多重继承程序示例	230
9.5.4 异常处理程序示例	233
9.6 小结	235
第 10 章 多线程编程	236
10.1 Linux 线程简介	236
10.1.1 线程的基本概念	236
10.1.2 线程的实现和创建	236
10.1.3 线程创建源代码	237
10.2 LinuxThreads 线程库	242
10.2.1 pthread_create ()	243
10.2.2 pthread_join ()	243
10.2.3 pthread_self ()	243
10.2.4 pthread_detach ()	244
10.2.5 pthread_exit ()	244
10.2.6 pthread_cancel ()	244
10.3 线程间通信	244
10.3.1 pthread_once ()	245
10.3.2 pthread_key_create ()	245
10.3.3 pthread_getspecific ()	246
10.3.4 pthread_setspecific ()	246
10.3.5 pthread_mutex_lock ()	246
10.3.6 pthread_mutex_unlock ()	246
10.3.7 pthread_cond_init ()	246
10.3.8 pthread_cond_wait ()	247
10.4 守护线程分析	247
10.5 多线程编程实例分析	249
10.5.1 X Window GUI 示例程序	249
10.5.2 TCP/IP 服务器示例程序	252
10.5.3 生产者和消费者模型示例程序	257

10.5.4 同步线程测试示例程序	258
10.5.5 同步线程读写缓冲区示例程序.....	259
10.6 小结	260

第 3 部分 建立嵌入式 Linux 系统

第 11 章 修改和编译 Linux 内核	263
11.1 概述	263
11.2 修改 Linux 内核示例	264
11.2.1 虚拟内存	264
11.2.2 实时调度	304
11.3 编译 Linux 内核	307
11.3.1 初始化环境	307
11.3.2 配置核心	307
11.3.3 编译内核	307
11.3.4 应用内核	307
11.4 小结	308
第 12 章 嵌入式 Linux 系统中的脚本控制	309
12.1 Linux 启动和退出过程简介	309
12.2 自定义决定运行级别的脚本	311
12.3 特定运行级别脚本	314
12.4 自定义公共启动脚本	316
12.5 自定义退出脚本	330
12.6 小结	334
第 13 章 组建嵌入式 Linux 系统的文件系统	335
13.1 概述	335
13.2 组建嵌入式 Linux 系统的文件系统脚本	335
13.3 分析嵌入式 Linux 系统的文件系统	337
13.4 小结	340

第 4 部分 嵌入式 Linux 系统的应用

第 14 章 IP 电话的实现原理	343
14.1 IP 电话应用结构	343
14.2 H.323 协议栈结构及应用	344
14.3 H.323 系统的组成	346

14.4 H.323 协议实体的功能	348
14.4.1 终端功能	348
14.4.2 网关功能	349
14.4.3 网守功能	350
14.4.4 多点通信端点的功能	351
14.5 相关协议简介	352
14.5.1 RAS 协议	352
14.5.2 H.255.0	353
14.5.3 H.245	355
14.5.4 RTP	356
14.5.5 RTCP	356
14.5.6 RSVP	357
14.6 小结	358
第 15 章 利用 PPCBOOT 引导系统	359
15.1 开发步骤简介	359
15.2 开发模型	360
15.3 修改 PPCBOOT 源代码	361
15.3.1 与主板有关的文件	361
15.3.2 与 CPU 有关的文件	362
15.3.3 头文件	364
15.3.4 公共函数文件	368
15.3.5 网络传输代码文件	371
15.3.6 Makefile 文件	375
15.4 PPCBoot 的监视界面	376
15.5 小结	377
第 16 章 建立开发平台	378
16.1 安装 CDK 1.2	378
16.2 设置目标系统	379
16.3 设置开发主机	379
16.3.1 设置 DHCP	380
16.3.2 设置 TFTP	380
16.3.3 设置 NFS	381
16.4 测试目标系统的内核装载与引导	381
16.5 开发平台的调试模型	383
16.5.1 开发应用	383
16.5.2 调试应用	384
16.6 小结	385

第 17 章 应用软件开发实例	386
17.1 需求分析.....	386
17.2 使用的协议栈和函数库.....	386
17.3 应用软件源代码.....	388
17.3.1 main.cxx.....	388
17.3.2 main.h.....	398
17.3.3 version.h.....	399
17.3.4 Makefile.....	399
17.4 小结.....	400
参考文献	401

第 1 部分 Linux 内核源代码分析

作为本书的开始，本部分简明扼要地介绍 Linux 的内核机制，为第 3 部分建立嵌入式 Linux 系统作铺垫。实际上，现在已有很多专门分析 Linux 内核的书籍，但本部分的介绍绝不是此类书籍的简单重复。作者在此深究 Linux 内核源代码只是精确分析与嵌入式 Linux 系统有关的内核机制。因此，本部分只重点介绍 Linux 的进程管理、系统调用、内存管理、文件系统和网络接口设备，而对 Linux 的模块加载、设备驱动等内容不作介绍。Linux 系统的进程管理包括进程创建、进程的调度以及进程间通信等，而 Linux 内核的内存管理则包括描述符、分页机制以及任务切换，它们是 Linux 内核分析的难点和重点。

第 1 章介绍 Linux 内核的进程管理，这里不仅有详尽的描述，还有核心代码文件的分析。理解嵌入式系统的进程管理，对于在该系统中创建进程项、进行多线程编程是十分有利的。实际上，不同的嵌入式系统，它们的进程管理机制大相径庭。例如，它们允许同时创建的进程个数就不一样，Linux 系统为 512 个。

第 2 章详细分析 Linux 内核的内存管理，即它的虚存管理机制。不同的操作系统管理内存的方法是不同的；即使同一操作系统，当它运行在不同的平台时它管理内存的方法也不一样。例如：Linux 系统在 X86 平台上使用二级分页机制，而在有些平台上使用三级分页机制。此外，为了满足实时性的要求，现在有些商家已经开发出了无虚拟内存的嵌入式 Linux 系统。

第 3 章和第 4 章对 Linux 内核的文件系统进行介绍。对于操作系统来说，文件系统也是它的一大特色之一。Windows 操作系统不同于 Linux 操作系统，也体现在文件系统的不一致上。当然，可以肯定，不同的嵌入式系统，它们的文件管理方法也不同。嵌入式 Linux 系统则使用 VFS 文件系统接口和 EXT2 文件系统。实际上，用户也可以在现有的 EXT2 文件系统上进行修改，在自己的嵌入式 Linux 系统中使用自定义的文件系统。

第 5 章简单介绍 Linux 内核的网络设备接口。与 Unix 系统内核一样，Linux 内核的网络通信是基于套接字 (socket) 进行的，它遵循 TCP/IP 协议。为什么把网络设备接口单列一章进行介绍呢？因为随着网络技术的发展，许多实际应用或多或少都与网络有一定联系，基于嵌入式 Linux 系统的应用也是如此。不仅这样，许多商业机构还制定了基于 TCP/IP 协议的应用协议。本书在第 3 部分介绍的应用实例就使用了基于 TCP/IP 协议的 H.323 协议族。

第 6 章讨论 Linux 内核的系统调用，有过 Linux 内核分析经验的用户已经知道，Linux 内核的系统调用不仅已用在内核的源代码中，而且它可以用在自己开发的应用程序中。正因为如此，对于开发嵌入式 Linux 系统的用户，应该理解 Linux 内核的系统调用原理，同时掌握 Linux 内核的系统调用接口。这对于在嵌入式 Linux 系统中开发应用程序是十分必要的。

