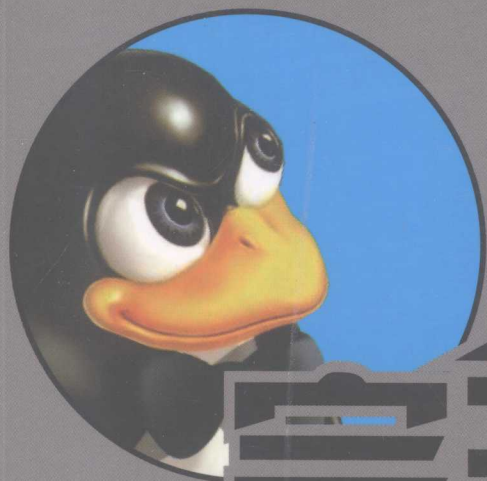


嵌入式Linux 应用开发



完全手册

- 基于ARM 9 + Linux 2.6平台
- 从基础讲起，引导读者快速入门
- 实例丰富，可直接应用于工程实践

本书全部实例代码及相关工具

韦东山 编著



嵌入式Linux 应用开发

完全手册

韦东山 编著



人民邮电出版社
北京

图书在版编目 (CIP) 数据

嵌入式 Linux 应用开发完全手册 / 韦东山编著. —北京:
人民邮电出版社, 2008.8
ISBN 978-7-115-18262-3

I. 嵌… II. 韦… III. Linux 操作系统—程序设计—技
术手册 IV. TP316.89-62

中国版本图书馆 CIP 数据核字 (2008) 第 082991 号

内 容 提 要

本书全面介绍了嵌入式 Linux 系统开发过程中, 从底层系统支持到上层 GUI 应用的方方面面, 内容涵盖 Linux 操作系统的安装及相关工具的使用、配置, 嵌入式编程所需要的基础知识 (交叉编译工具的选项设置、Makefile 语法、ARM 汇编指令等), 硬件部件的使用及编程 (囊括了常见硬件, 比如 UART、I²C、LCD 等), U-Boot、Linux 内核的分析、配置和移植, 根文件系统的构造 (包括移植 busybox、glibc、制作映象文件等), 内核调试技术 (比如添加 kgdb 补丁、栈回溯等), 驱动程序编写及移植 (LED、按键、扩展串口、网卡、硬盘、SD 卡、LCD 和 USB 等), GUI 系统的移植 (包含两个 GUI 系统: 基于 Qtopia 和基于 X), 应用程序调试技术。

本书从最简单的点亮一个 LED 开始, 由浅入深地讲解, 使读者最终可以配置、移植、裁剪内核, 编写驱动程序, 移植 GUI 系统, 掌握整个嵌入式 Linux 系统的开发方法。

本书由浅入深, 循序渐进, 适合刚接触嵌入式 Linux 的初学者学习, 也可作为大、中专院校嵌入式相关专业本科生、研究生的教材。

嵌入式 Linux 应用开发完全手册

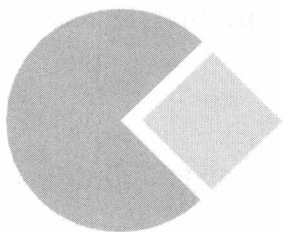
- ◆ 编 著 韦东山
责任编辑 黄 焱
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京昌平百善印刷厂印刷
- ◆ 开本: 787×1092 1/16
印张: 37.25
字数: 908 千字 2008 年 8 月第 1 版
印数: 1—4 000 册 2008 年 8 月北京第 1 次印刷

ISBN 978-7-115-18262-3/TP

定价: 69.00 元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154



前言

背景知识

嵌入式 Linux 在嵌入式领域发展迅速、需求旺盛，但是嵌入式 Linux 的入门很难。初学者多是自己琢磨，效率不高。学习过程中碰到的问题千奇百怪，解决后却往往发现是极其低级的错误，以作者为例，初学时在论坛疯狂发帖求教，现在回头一看不免感叹：怎么会提出这么弱智的问题？但是，当时就是被这类问题折磨得寝食难安。

相对于嵌入式 Linux 常识的匮乏，更大的困难是缺乏完善的知识结构：只了解硬件，或是只了解软件。对于有志于从事底层系统开发（比如改造 Bootloader、钻研内核、为新硬件编写驱动程序）的人，对于想从上层软件开发转到底层软件开发的人，应该看得懂电路原理图，看得懂芯片数据手册，清楚地知道软件是怎样和硬件发生作用的。

同样，对于想从硬件岗位转到软件岗位的人，对于想从传统单片机（比如 51 单片机）编程进一步学习“有操作系统的”嵌入式编程的人，需要找到一个学习的切入点：先掌握各个硬件部件的简单编程，再将它们组合起来构成一个相对复杂的软件系统——比如 Bootloader，进而编写基于操作系统的驱动程序，最后深入钻研操作系统内核。

对于尚未参加工作的在校生来说，缺乏实际的操作经验可能是就业的最大障碍。很多人买了开发板想进一步练习，却发现不知从何入手。

鉴于上述种种困难及需求，作者结合自己的学习经历、工作心得写成此书，期望能帮助读者加快嵌入式 Linux 的入门速度，并体会到深入学习嵌入式 Linux 的乐趣。

关于本书

本书以 S3C2410、S3C2440 开发板为例，从分析硬件上电执行的第一条指令开始，到构造出一个类似 PDA、基于 Linux 的桌面 GUI 系统，带领读者学习、掌握从最底层到最高层的软件编写方法。

本书主要涉及以下主题：

- 开发环境的搭建（包括安装 Linux 系统及日常使用的工具）；
- 开发板上各硬件部件的使用方法及实际的编程操作；

- 嵌入式 Linux 系统的构造（包括 Bootloader、内核、文件系统等）；
- 嵌入式 Linux 驱动程序的编写方法及大量实例；
- GUI 系统的移植（两个 GUI 系统：基于 Qtopia 和基于 X）；
- 调试技术（包括内核调试技术和应用程序调试技术）。

本书所有章节都以理论结合代码的方式进行讲解，并可按照书中说明进行实际操作，力求让读者“知其然，也知其所以然”。

本书内容及组织方式

本书按照嵌入式 Linux 初学者的学习过程，从简单到复杂，从底层软件到上层软件进行讲解，全书分 5 篇，共 27 章。

第 1 篇（第 1 章至第 4 章）为嵌入式 Linux 开发环境构建篇，主要讲解以下内容。

- 第 1 章介绍基于 ARM 的嵌入式 Linux 系统的基本概念。
- 第 2 章讲解嵌入式开发环境的建立，包括在 PC 上安装、配置 Linux 操作系统，安装随书光盘。
- 第 3 章介绍交叉编译工具的选项、Makefile 的语法以及本书用到的 ARM 汇编指令及相关知识，这章可以当作阅读后续章节时的参考手册。
- 第 4 章介绍了一些日常工作要用到工具，比如源码阅读、编辑工具等。

第 2 篇（第 5 章至第 14 章）为 ARM9 嵌入式系统基础实例篇，具体内容如下。

本篇首先根据 S3C2410、S3C2440 的数据手册介绍各硬件部件的使用方法，然后介绍怎样编写程序来操作它们。文中穿插介绍了连接器的很多使用技巧，读者可以由此接触到“程序的内部结构”，这是单纯的上层开发人员所缺乏的。通过读写各个硬件部件的寄存器来操作硬件，读者还可以深刻体会到“软件”和“硬件”是怎样发生作用的，是第 3 篇、第 4 篇的基础。

第 3 篇（第 15 章至第 18 章）为嵌入式 Linux 系统移植篇，主要讲解以下内容。

- 第 15 章深入分析 U-Boot（它负责引导内核）的代码结构，并详细介绍了将它移植到开发板上的方法。
- 第 16 章首先分析了内核的代码结构，然后深入分析它的启动过程，最后将它移植到开发板上。
- 第 17 章先从整体上介绍了 Linux 文件系统的目录结构——FHS 标准。然后构造文件系统：移植常用工具的集合 Busybox，移植 glibc 库，建立各个目录，建立配置文件。最后修改、编译一些工具，使用它们来制作 yaffs、jffs2 文件系统映像文件。
- 第 18 章介绍了 3 种内核调试技术：printk、kgdb 补丁、使用 Oops 信息进行栈回溯。

第 4 篇（第 19 章至第 24 章）为嵌入式 Linux 设备驱动开发篇，具体内容如下。

在第 19 章中总体介绍了驱动程序的编写、移植方法，在第 20 章介绍了内核的异常处理体系结构——就是怎样使用中断。

其他章节都是一些例子：先总体介绍相关硬件的驱动程序架构，然后根据开发板的特性进行修改。

第 5 篇（第 25 章至第 27 章）为嵌入式 Linux 系统应用开发篇，主要讲解以下内容。

- 第 25 章移植了一个基于 Qtopia 的 GUI 系统，并且以简单的“Hello, world”程序为例编写、调试 GUI 程序。

- 第 26 章移植了一个基于 X 的 GUI 系统, 里面涉及众多软件, 读者可以体会到上层应用的开发过程, 并且获得移植大型软件的经验。这章还介绍了一个名为 Scratchbox 的交叉编译工具包, 它虚拟出一个可以直接编译软件的目标机器, 使得“交叉编译”变为“本地编译”, 大幅减少了为非 x86 平台移植软件所需的工作量。
- 第 27 章介绍了几种简便的应用程序调试技术, 包括使用 strace 工具跟踪系统调用和信号, 使用 memwatch 检查程序的内存漏洞, 使用库函数 backtrace 和 backtrace_symbols 来定位段错误。

本书特色

- 由浅入深, 从最简单的点亮 LED 讲起直至移植 GUI 系统。
- 实例丰富, 每个实例都详尽地介绍原理及分析代码。
- 结构合理, 先总体介绍概念、架构, 然后进行具体操作。
- 包括初学者所碰到的常见问题。

参与本书编写的人员

本书由韦东山负责编写并统编全部书稿, 陈汉仪、于明俭对本书的写作提供了大力支持, 在此表示感谢。

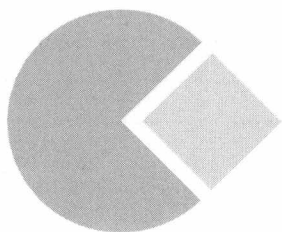
感谢我的父母和女友, 在本书写作过程中给了我强大的精神支持, 鼓励、支持我, 使我能够坚持写完本书。

同时参与编写的还有柴作朋、单辉、丁鹏、冯发勇、付贤会、葛仕明、何国宝、何圆明、何化成、黄永华、李志宏、廖娟、林清妹、陆江萍、祁晓璐、谭爱华、魏明辉、张帮芹、周霜、朱旭琪等, 在此一并表示感谢。

我们为本书开通了专用的网站, 网址是 <http://www.100ask.net>, 读者可以直接同我们交流, 共同学习和提高。

由于水平有限, 书中难免遗漏和不足之处, 恳请广大读者提出宝贵意见。本书责任编辑的联系方式是 huangyan@ptpress.com.cn, 欢迎来信交流。

编 者
2008 年 6 月



目 录

第 1 篇 嵌入式 Linux 开发环境构建篇

第 1 章 嵌入式 Linux 开发概述	2
1.1 嵌入式系统介绍	2
1.1.1 嵌入式系统的定义和特点	2
1.1.2 嵌入式技术的发展历史	3
1.2 基于 ARM 处理器的嵌入式 Linux 系统	5
1.2.1 ARM 处理器介绍	5
1.2.2 在嵌入式系统中选择嵌入式 Linux 的理由	8
第 2 章 嵌入式 Linux 开发环境构建	10
2.1 硬件环境构建	10
2.1.1 主机与目标板结合的交叉开发模式	10
2.1.2 硬件要求	11
2.2 软件环境构建	12
2.2.1 主机 Linux 操作系统的安装	12
2.2.2 主机 Linux 操作系统上网络服务的配置与启动	18
2.2.3 在主机 Linux 操作系统中安装基本的开发环境	23
2.2.4 光盘的内容结构及安装	23
2.2.5 安装交叉编译工具链	25
2.2.6 书中写作风格的约定	28
第 3 章 嵌入式编程基础知识	29
3.1 交叉编译工具选项说明	29
3.1.1 arm-linux-gcc 选项	29
3.1.2 arm-linux-ld 选项	38

3.1.3	arm-linux-objcopy 选项	41
3.1.4	arm-linux-objdump 选项	43
3.1.5	汇编代码、机器码和存储器的关系以及数据的表示	44
3.2	Makefile 介绍	45
3.2.1	Makefile 规则	45
3.2.2	Makefile 文件里的赋值方法	46
3.2.3	Makefile 常用函数	46
3.3	常用 ARM 汇编指令及 ATPCS 规则	52
3.3.1	本书使用的所有汇编指令	52
3.3.2	ARM-THUMB 子程序调用规则 ATPCS	55
第 4 章	Windows、Linux 环境下相关工具、命令的使用	58
4.1	Windows 环境下的工具介绍	58
4.1.1	代码阅读、编辑工具 Source Insight	58
4.1.2	文件传输工具 Cuteftp	63
4.1.3	远程登录工具 SecureCRT	63
4.1.4	TFTP 服务器软件 Tftpd32	64
4.2	Linux 环境下的工具、命令介绍	65
4.2.1	代码阅读、编辑工具 KScope	65
4.2.2	远程登录工具 C-kernel	69
4.2.3	编辑命令 vi	69
4.2.4	查找命令 grep、find 命令	71
4.2.5	在线手册查看命令 man	72
4.2.6	其他命令: tar、diff、patch	73
第 2 篇 ARM9 嵌入式系统基础实例篇		
第 5 章	GPIO 接口	76
5.1	GPIO 硬件介绍	76
5.1.1	通过寄存器来操作 GPIO 引脚	76
5.1.2	怎样使用软件来访问硬件	77
5.2	GPIO 操作实例: LED 和按键	80
5.2.1	硬件设计	80
5.2.2	程序设计及代码详解	80
5.2.3	实例测试	86
第 6 章	存储器控制	87
6.1	使用存储控制器访问外设的原理	87

6.1.1	S3C2410/S3C2440 的地址空间	87
6.1.2	存储控制器与外设的关系	89
6.1.3	存储控制器的寄存器使用方法	91
6.2	存储控制器操作实例：使用 SDRAM	94
6.2.1	代码详解及程序的复制、跳转过程	94
6.2.2	实例测试	97
第 7 章	内存管理单元 MMU	98
7.1	内存管理单元 MMU 介绍	98
7.1.1	S3C2410/S3C2440 MMU 特性	98
7.1.2	S3C2410/S3C2440 MMU 地址变换过程	99
7.1.3	内存的访问权限检查	107
7.1.4	TLB 的作用	109
7.1.5	Cache 的作用	110
7.1.6	S3C2410/S3C2440 MMU、TLB、Cache 的控制指令	113
7.2	MMU 使用实例：地址映射	113
7.2.1	程序设计	113
7.2.2	代码详解	114
7.2.3	实例测试	124
第 8 章	NAND Flash 控制器	125
8.1	NAND Flash 介绍和 NAND Flash 控制器使用	125
8.1.1	Flash 介绍	125
8.1.2	NAND Flash 的物理结构	127
8.1.3	NAND Flash 访问方法	128
8.1.4	S3C2410/S3C2440 NAND Flash 控制器介绍	134
8.2	NAND Flash 控制器操作实例：读 Flash	135
8.2.1	读 NAND Flash 的步骤	135
8.2.2	代码详解	137
第 9 章	中断体系结构	143
9.1	S3C2410/S3C2440 中断体系结构	143
9.1.1	ARM 体系 CPU 的 7 种工作模式	143
9.1.2	S3C2410/S3C2440 中断控制器	146
9.1.3	中断控制器寄存器	149
9.2	中断控制器操作实例：外部中断	151
9.2.1	按键中断代码详解	151
9.2.2	实例测试	158

第 10 章 系统时钟和定时器	159
10.1 时钟体系及各类时钟部件	159
10.1.1 S3C2410/S3C2440 时钟体系	159
10.1.2 PWM 定时器	161
10.1.3 WATCHDOG 定时器	164
10.2 MPLL 和定时器操作实例	166
10.2.1 程序设计	166
10.2.2 代码详解	166
10.2.3 实例测试	170
第 11 章 通用异步收发器 UART	171
11.1 UART 原理及 UART 部件使用方法	171
11.1.1 UART 原理说明	171
11.1.2 S3C2410/S3C2440 UART 的特性	172
11.1.3 S3C2410/S3C2440 UART 的使用	173
11.2 UART 操作实例	177
11.2.1 代码详解	177
11.2.2 实例测试	180
第 12 章 I²C 接口	181
12.1 I ² C 总线协议及硬件介绍	181
12.1.1 I ² C 总线协议	181
12.1.2 S3C2410/S3C2440 I ² C 总线控制器	184
12.2 I ² C 总线操作实例	187
12.2.1 I ² C 接口 RTC 芯片 M41t11 的操作方法	187
12.2.2 程序设计	188
12.2.3 设置/读取 M41t11 的源码详解	188
12.2.4 I ² C 实例的连接脚本	195
12.2.5 实例测试	196
第 13 章 LCD 控制器	197
13.1 LCD 和 LCD 控制器	197
13.1.1 LCD 显示器	197
13.1.2 S3C2410/S3C2440 LCD 控制器介绍	199
13.2 TFT LCD 显示实例	210
13.2.1 程序设计	210
13.2.2 代码详解	210
13.2.3 实例测试	221

第 14 章 ADC 和触摸屏接口	222
14.1 ADC 和触摸屏硬件介绍及使用	222
14.1.1 S3C2410/S3C2440 ADC 和触摸屏接口概述	222
14.1.2 S3C3410/S3C2440 ADC 接口的使用方法	224
14.1.3 触摸屏原理及接口	226
14.2 ADC 和触摸屏操作实例	230
14.2.1 硬件设计	230
14.2.2 程序设计	230
14.2.3 测试 ADC 的代码详解	230
14.2.4 测试触摸屏的代码详解	232
14.2.5 实例测试	237

第 3 篇 嵌入式 Linux 系统移植篇

第 15 章 移植 U-Boot	240
15.1 Bootloader 简介	240
15.1.1 Bootloader 的概念	240
15.1.2 Bootloader 的结构和启动过程	241
15.1.3 常用 Bootloader 介绍	246
15.2 U-Boot 分析与移植	246
15.2.1 U-Boot 工程简介	246
15.2.2 U-Boot 源码结构	247
15.2.3 U-Boot 的配置、编译、连接过程	249
15.2.4 U-Boot 的启动过程源码分析	257
15.2.5 U-Boot 的移植	264
15.2.6 U-Boot 的常用命令	288
15.2.7 使用 U-Boot 来执行程序	292
第 16 章 移植 Linux 内核	293
16.1 Linux 版本及特点	293
16.2 Linux 移植准备	294
16.2.1 获取内核源码	294
16.2.2 内核源码结构及 Makefile 分析	295
16.2.3 内核的 Kconfig 分析	304
16.2.4 Linux 内核配置选项	309
16.3 Linux 内核移植	313

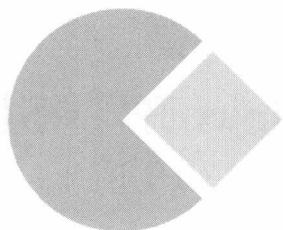
16.3.1	Linux 内核启动过程概述	313
16.3.2	修改内核以支持 S3C2410/S3C2440 开发板	314
16.3.3	修改 MTD 分区	327
16.3.4	移植 YAFFS 文件系统	330
16.3.5	编译、烧写、启动内核	333
第 17 章 构建 Linux 根文件系统		335
17.1	Linux 文件系统概述	335
17.1.1	Linux 文件系统的特点	335
17.1.2	Linux 根文件系统目录结构	336
17.1.3	Linux 文件属性介绍	340
17.2	移植 Busybox	341
17.2.1	Busybox 概述	341
17.2.2	init 进程介绍及用户程序启动过程	342
17.2.3	编译/安装 Busybox	346
17.3	使用 glibc 库	350
17.3.1	glibc 库的组成	350
17.3.2	安装 glibc 库	351
17.4	构建根文件系统	352
17.4.1	构建 etc 目录	352
17.4.2	构建 dev 目录	354
17.4.3	构建其他目录	356
17.4.4	制作/使用 yaffs 文件系统映象文件	356
17.4.5	制作/使用 jffs2 文件系统映象文件	360
第 18 章 Linux 内核调试技术		362
18.1	内核打印函数 printk	362
18.1.1	printk 的使用	362
18.1.2	串口控制台	364
18.2	内核源码级别的调试方法	366
18.2.1	内核调试工具 KGDB 的作用与原理	366
18.2.2	给内核添加 KGDB 功能支持 S3C2410/S3C2440	367
18.2.3	结合可视化图形前端 DDD 和 gdb 来调试内核	372
18.3	Oops 信息及栈回溯	375
18.3.1	Oops 信息来源及格式	375
18.3.2	配置内核使 Oops 信息的栈回溯信息更直观	376
18.3.3	使用 Oops 信息调试内核的实例	376
18.3.4	使用 Oops 的栈信息手工进行栈回溯	380

第 4 篇 嵌入式 Linux 设备驱动开发篇

第 19 章 字符设备驱动程序	384
19.1 Linux 驱动程序开发概述	384
19.1.1 应用程序、库、内核、驱动程序的关系	384
19.1.2 Linux 驱动程序的分类和开发步骤	385
19.1.3 驱动程序的加载和卸载	387
19.2 字符设备驱动程序开发	387
19.2.1 字符设备驱动程序中重要的数据结构和函数	387
19.2.2 LED 驱动程序源码分析	389
第 20 章 Linux 异常处理体系结构	396
20.1 Linux 异常处理体系结构概述	396
20.1.1 Linux 异常处理的层次结构	396
20.1.2 常见的异常	400
20.2 Linux 中断处理体系结构	401
20.2.1 中断处理体系结构的初始化	401
20.2.2 用户注册中断处理函数的过程	404
20.2.3 中断的处理过程	406
20.2.4 卸载中断处理函数	409
20.3 使用中断的驱动程序示例	410
20.3.1 按键驱动程序源码分析	410
20.3.2 测试程序情景分析	415
第 21 章 扩展串口驱动程序移植	419
21.1 串口驱动程序框架概述	419
21.1.1 串口驱动程序术语介绍	419
21.1.2 串口驱动程序的 4 层结构	420
21.2 扩展串口驱动程序移植	423
21.2.1 串口驱动程序低层代码分析	423
21.2.2 修改代码以支持扩展串口	425
21.2.3 测试扩展串口	429
第 22 章 网卡驱动程序移植	431
22.1 CS8900A 网卡驱动程序移植	431
22.1.1 CS8900A 网卡特性	431
22.1.2 CS8900A 网卡驱动程序修改	432

22.2	DM9000 网卡驱动程序移植	441
22.2.1	DM9000 网卡特性	441
22.2.2	DM9000 网卡驱动程序修改	442
第 23 章	IDE 接口和 SD 卡驱动程序移植	450
23.1	IDE 接口驱动程序移植	450
23.1.1	IDE 接口相关概念介绍	450
23.1.2	IDE 接口驱动程序移植	452
23.1.3	IDE 接口驱动程序测试	461
23.2	SD 卡驱动程序移植	464
23.2.1	SD 卡相关概念介绍	464
23.2.2	SD 卡驱动程序移植	465
23.2.3	SD 卡驱动程序测试	472
23.2.4	磁盘分区表	473
第 24 章	LCD 和 USB 驱动程序移植	475
24.1	LCD 驱动程序移植	475
24.1.1	LCD 和 USB 键盘驱动程序框架	475
24.1.2	S3C2410/S3C2440 LCD 控制器驱动程序移植	479
24.2	USB 驱动程序移植	489
24.2.1	USB 驱动程序概述	489
24.2.2	配置内核支持 USB 键盘、USB 鼠标和 USB 硬盘	491
24.2.3	USB 设备的使用	492
第 5 篇 嵌入式 Linux 系统应用开发篇		
第 25 章	基于 Qtopia 的 GUI 开发	496
25.1	嵌入式 GUI 介绍	496
25.1.1	Linux 桌面 GUI 系统的发展	496
25.1.2	嵌入式 Linux 中的几种 GUI	499
25.2	Qtopia 移植	501
25.2.1	主机开发环境的搭建	501
25.2.2	交叉编译、安装 Qtopia 2.2.0	502
25.2.3	开发自己的 Qt GUI 程序	514
25.2.4	在主机上使用模拟软件开发、调试嵌入式 Qt GUI 程序	518
第 26 章	基于 X 的 GUI 开发	524
26.1	X Window 概述	524

26.1.1	X 协议介绍	524
26.1.2	窗口管理器 (Window manager)	526
26.1.3	桌面环境 (Desktop environment)	526
26.2	交叉编译工具包 Scratchbox	526
26.2.1	Scratchbox 介绍	527
26.2.2	安装 Scratchbox 及编译工具	528
26.2.3	在 Scratchbox 里安装交叉编译工具链	529
26.2.4	安装其他开发工具	535
26.3	移植 X	536
26.3.1	编译软件的基本知识	536
26.3.2	编译 X 的依赖软件	539
26.3.3	编译 Xorg	542
26.4	移植 Matchbox	547
26.4.1	下载源代码	548
26.4.2	编译 Matchbox	548
26.4.3	运行、试验 Matchbox	550
26.5	移植 GTK+	553
26.5.1	GTK+介绍	553
26.5.2	GTK+移植	553
26.6	移植基于 GTK+/X 的 GUI 程序	555
26.6.1	xterm 移植	556
26.6.2	gtkboard 移植	557
26.6.3	裁剪文件系统	560
第 27 章	Linux 应用程序调试技术	564
27.1	使用 strace 工具跟踪系统调用和信号	564
27.1.1	strace 介绍及移植	564
27.1.2	使用 strace 来调试程序	565
27.2	内存调试工具	568
27.2.1	使用 memwatch 进行内存调试	568
27.2.2	其他内存工具介绍: mtrace、dmalloc、yamd	571
27.3	段错误的调试方法	573
27.3.1	使用库函数 backtrace 和 backtrace_symbols 定位段错误	573
27.3.2	段错误调试实例	574
参考文献		578



第 1 篇 嵌入式 Linux 开发环境构建篇

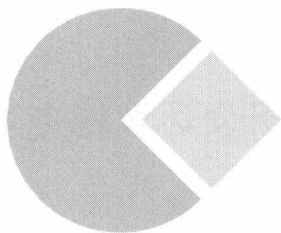
嵌入式 Linux 开发概述 ■

嵌入式 Linux 开发环境构建 ■

嵌入式编程基础知识 ■

Windows、Linux 环境下相关工具、命令的使用 ■

.....



第1章 嵌入式 Linux 开发概述

本章目标

- 了解嵌入式系统的概念及发展历史
- 了解 ARM 处理器
- 了解各类嵌入式操作系统

1.1 嵌入式系统介绍

1.1.1 嵌入式系统的定义和特点

1. 嵌入式系统的定义

嵌入式系统的定义为：以应用为中心、以计算机技术为基础、软硬件可裁剪、适用于应用系统，对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。它的主要特点是嵌入、专用。

从 20 世纪 70 年代起，微型机以小型、价廉、高速数值计算等特点迅速走向市场，它所具备的智能化水平在工业控制领域发挥了作用，常被组装成各种形状，“嵌入”到一个对象体系中，进行某类智能化的控制。这样一来，计算机便失去了原来的形态与“通用”的功能，为区别于通用计算机系统，将这类为了某个“专用”的目的，而“嵌入”到对象体系中的计算机系统，称为嵌入式计算机系统，简称嵌入式系统。

含有嵌入式系统的设备称为嵌入式设备，这在生活中随处可见：电子表、手机、MP3 播放器、摇控器等，涵盖了生产、工业控制、通信、网络、消费电子、汽车电子、军工等领域。从通俗、广义的角度来说，除电脑、超级计算机等具备比较强大计算能力及系统资源（比如内存、存储器等）的电子系统之外，凡具备计算能力的设备都可称为嵌入式设备。随着技术的进步，嵌入式设备的性能越来越高，一个相对高级的 PDA 的性能并不弱于一般的电脑。

2. 嵌入式系统的特点

嵌入式设备常应用于“特定”场合，与“通用的”个人电脑相比，具备以下特点。