

从实践中学

嵌入式Linux应用程序开发

第2版

华清远见嵌入式学院 苗德行 冯建 刘洪涛 潘启勇 编著



本书特色:

凝聚业内著名讲师一线教学经验
汇总百家知名企业最新人才标准
精选实用案例直击真实项目需求
结合学习思路提供配套技术资料

`int gpio_open (struct inode *inode, struct file *filp) Intent intent = new Intent(Main.this, Seco...`



中国工信出版集团



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



高等院校嵌入式人才培养规划教材

从实践中学

嵌入式Linux应用程序开发

第2版

华清远见嵌入式学院 苗德行 冯建 刘洪涛 潘启勇 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书结合大量实例,讲解了嵌入式 Linux 应用程序设计各个方面的基本方法及必要的核心概念。主要内容包括搭建嵌入式 Linux 开发环境、嵌入式文件 I/O 编程、嵌入式 Linux 多任务编程、嵌入式 Linux 进程间通信、嵌入式 Linux 多线程编程、嵌入式 Linux 网络编程、嵌入式 Linux 设备驱动编程、Android 应用编程、Android 播放器项目设计等。重视应用是贯穿全书的最大特点,在各章和全书结尾分别设置了在项目实践中常见或类似的应用实例。

本书可作为大学院校电子、通信、计算机、自动化等专业的嵌入式 Linux 开发课程的教材,也可供嵌入式开发人员参考。学习本书应具有 Linux C 语言编程和 Java 编程的基本知识。

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

从实践中学嵌入式 Linux 应用程序开发 / 华清远见嵌入式学院等编著. —2 版. —北京:电子工业出版社, 2015.8

高等院校嵌入式人才培养规划教材

ISBN 978-7-121-26471-9

I. ①从… II. ①华… III. ①Linux 操作系统—程序设计—高等学校—教材 IV. ①TP316.89

中国版本图书馆 CIP 数据核字(2015)第 145107 号

策划编辑:孙学瑛

责任编辑:徐津平

特约编辑:赵树刚

印 刷:三河市双峰印刷装订有限公司

装 订:三河市双峰印刷装订有限公司

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:22 字数:525 千字

版 次:2012 年 3 月第 1 版

2015 年 8 月第 2 版

印 次:2015 年 8 月第 1 次印刷

定 价:59.00 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。

推荐序

移动与云计算的发展推动了越来越多的新技术、新应用和新产品的涌现，推动了嵌入式电子产品世界的不断更新和快速发展。作为嵌入式行业最著名的厂商之一，20多年来 ARM 除了不断地加大研发投入，开发最新的微处理器、图形技术、物理 IP 和开发工具，为产业升级搭建了最佳的开发架构；同时，也一直致力于建设一个开放的、具有强大生命力和发展前景的 ARM 嵌入式生态系统，使得每个存在于这个生态系统的成员都能发挥各自的特长，通过有效的产业分工和协作开发出高性能、低功耗、人性化的嵌入式产品服务于广大的消费者。

在这个生态系统中，嵌入式操作系统是必不可少的重要环节，是链接底层硬件和上层应用软件的纽带。其中，Linux 作为开源的嵌入式操作系统，多年来一直受到广大工程师朋友的喜爱，特别是在基于 Linux 内核的 Android 操作系统发布以来，Linux 的应用和发展到了一个崭新的高度。ARM 作为应用最广泛的嵌入式处理器，对 Linux 操作系统的发展也给予了大量的支持与贡献。

吴雄昂
ARM 中国区总经理

前 言

在今天所处的大时代背景下，嵌入式、3G、物联网、云计算俨然已经成为信息产业的主旋律，不管是从政府大力扶持，还是从产业变革来说，这股潮流早已势不可当。而嵌入式系统正是这些产业应用技术中最核心的部分。随着智能化电子行业的迅猛发展，嵌入式行业更是凭借其“应用领域广、人才需求大、就业薪资高、行业前景好”等众多优势，成为当前最热门、最有发展前途的行业之一，与此同时，嵌入式研发工程师更是成为 IT 职场的紧缺人才。因此，近几年来，各院校纷纷开设嵌入式专业课程。但是，各院校在嵌入式专业教学建设的过程中几乎都面临教材难觅、内容更新迟缓的困境。虽然目前市场上嵌入式开发相关书籍比较多，但几乎都是针对有一定基础的行业内研发人员而编写的，并不完全符合高校的教学要求。

针对高校专业教材缺乏的现状，我们以多年来在嵌入式工程技术领域内人才培养、项目研发的经验为基础，汇总了近几年积累的数百家企业对嵌入式研发相关岗位的真实需求，并结合行业应用技术的最新状况及未来发展趋势，调研了数十所开设“嵌入式工程技术”专业的院校的课程设置情况、学生特点和教学用书现状。通过细致地整理和分析，对专业技能和基本知识进行合理划分，编写了这套高等院校嵌入式人才培养规划教材，包括：

- 《从实践中学 ARM 体系结构与接口技术》
- 《从实践中学嵌入式 Linux 操作系统》
- 《从实践中学嵌入式 Linux C 编程》
- 《从实践中学嵌入式 Linux 应用程序开发（第 2 版）》

本套教材按照专业整体教学要求组织编写，各自对应的主干课程之间既相对独立又有机衔接，整套教材具有系统性。《从实践中学 ARM 体系结构与接口技术》侧重介绍接口技术；在操作系统教材方面，根据各院校的教学重点和行业实际应用情况，编写了《从实践中学嵌入式 Linux 操作系统》；考虑到嵌入式专业对学生 C 语言能力要求较高，编写了《从实践中学嵌入式 Linux C 编程》，可作为“C 语言基础”课程的后续提高课程使用；《从实践中学嵌入式 Linux 应用程序开发（第 2 版）》则重点突出了贯穿前面所学知识的实训内容，供“嵌入式 Linux 应用开发”课程使用。

书中结合大量代码和实例，循序渐进地讲解了嵌入式 Linux 应用软件开发的核心技能、经验和技巧。

全书共 9 章。前 7 章是对 Linux 环境下应用开发方法的学习，各章包含相应的实验内容；第 8 章是 Android 应用编程的基础知识；第 9 章安排了一个 Android 应用的实训内容。

第1章为搭建嵌入式Linux开发环境，首先介绍了交叉编译环境等嵌入式开发环境的搭建，然后讲解了嵌入式Linux系统中Bootloader、内核、文件系统的构建方法。

第2章为嵌入式文件I/O编程，主要讲解了Linux系统调用、Linux文件I/O系统、底层文件I/O操作、嵌入式Linux串口应用编程、标准I/O编程等内容。

第3章为嵌入式Linux多任务编程，主要讲解了Linux环境下的进程控制方法。

第4章为嵌入式Linux进程间通信，主要讲解了几种常用的进程间通信方法，包括管道通信、信号通信、信号量、共享内存、消息队列等。

第5章为嵌入式Linux多线程编程，主要讲解了Linux环境下的多线程编程方法及注意事项。

第6章为嵌入式Linux网络编程，主要讲解了Linux环境下的网络编程方法，涉及网络的非阻塞访问、异步处理、多路复用等。

第7章为嵌入式Linux设备驱动编程，主要介绍了Linux设备驱动编程基础和字符设备驱动编程的基本思路，并介绍了在S3C2410开发平台上编写GPIO驱动和按键驱动程序的基本思路。

第8章为Android应用编程，包括Android应用开发环境搭建、Android应用图形界面设计基础、Android主要组件等。

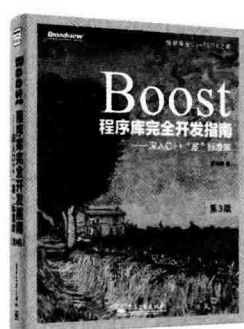
第9章为Android播放器项目设计，以一个简易的Android播放器为例，将Android应用开发所需掌握的相关知识贯穿起来，让读者能对Android应用开发有一个全局的视野。

本书由华清远见嵌入式学院资深讲师苗德行、冯建、刘洪涛、潘启勇编著并统校全稿。还要感谢华清远见嵌入式学院，教材内容参考了学院与嵌入式企业需求无缝对接的、科学的专业人才培养体系。同时，嵌入式学院从业或执教多年的行业专家团队也对教材的编写工作作出了贡献，张志华、蔡蒙、王利丽、张丹、杨曼、谭翠君、关晓强、李媛媛、卢闫进、赵松、邱迎龙和贾燕枫等教师在书稿的编写过程中认真阅读了所有章节，提供了大量在实际教学中积累的重要素材，对教材结构、内容提出了中肯的建议，并在后期审校工作中提供了很多帮助，在此表示衷心的感谢。

由于编者水平所限，书中难免存在不妥之处，恳请读者批评指正。对于本书的批评和建议，可以发到www.embedu.org技术论坛。

编 者
2015年5月

电子工业出版社计算机精品图书



反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010)88254396；(010)88258888

传 真：(010)88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱 电子工业出版社总编办公室

邮 编：100036

目 录

第 1 章 搭建嵌入式 Linux 开发环境	1
1.1 构建嵌入式 Linux 交叉开发环境	2
1.1.1 搭建嵌入式交叉编译环境	2
1.1.2 配置主机交叉开发环境	6
1.2 Bootloader	10
1.2.1 Bootloader 的种类	10
1.2.2 U-Boot 编译与使用	12
1.2.3 U-Boot 移植	24
1.3 Linux 内核与移植	27
1.3.1 Linux 内核结构	28
1.3.2 Linux 内核配置与编译	29
1.3.3 Linux 内核移植的简介	32
1.4 嵌入式文件系统构建	33
1.5 本章小结	37
1.6 本章习题	37
第 2 章 嵌入式文件 I/O 编程	38
2.1 Linux 系统调用及用户编程接口 (API)	39
2.1.1 Linux 系统调用	39
2.1.2 用户编程接口 (API)	39
2.1.3 系统命令	40
2.2 Linux 文件 I/O 系统概述	40
2.2.1 虚拟文件系统 (VFS)	40
2.2.2 Linux 中的文件及文件描述符	41
2.3 底层文件 I/O 操作	42
2.3.1 基本文件操作	42
2.3.2 文件锁	46

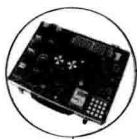
2.3.3 多路复用	51
2.4 嵌入式 Linux 串口应用编程.....	57
2.4.1 串口编程基础知识	57
2.4.2 串口配置	58
2.4.3 串口使用	67
2.5 标准 I/O 编程.....	70
2.6 实验内容.....	79
2.6.1 文件读写及上锁	79
2.6.2 多路复用式串口操作	85
2.7 本章小结	89
2.8 本章习题	90
第 3 章 嵌入式 Linux 多任务编程	91
3.1 Linux 下多任务机制的介绍.....	92
3.1.1 任务	92
3.1.2 进程	92
3.1.3 线程	98
3.2 进程控制编程.....	99
3.2.1 进程编程基础	99
3.2.2 Linux 守护进程.....	110
3.2.3 Linux 僵尸进程.....	117
3.3 实验内容.....	123
3.3.1 编写多进程程序	123
3.3.2 编写守护进程	127
3.4 本章小结	129
3.5 本章习题	130
第 4 章 嵌入式 Linux 进程间通信	131
4.1 Linux 下进程间通信概述.....	132
4.2 管道通信	133
4.2.1 管道简介	133
4.2.2 无名管道系统调用	134
4.2.3 标准流管道	137



从实践中学嵌入式 Linux 应用程序开发(第2版)

4.2.4 有名管道 (FIFO)	139
4.3 信号通信	142
4.3.1 信号概述	142
4.3.2 信号发送与捕捉	144
4.4 信号量	153
4.4.1 信号量概述	153
4.4.2 信号量编程	154
4.5 共享内存	158
4.6 消息队列	163
4.7 实验内容	168
4.7.1 有名管道通信实验	168
4.7.2 共享内存实验	172
4.8 本章小结	176
4.9 本章习题	177
第5章 嵌入式 Linux 多线程编程	178
5.1 多线程编程	179
5.1.1 线程基本编程	179
5.1.2 线程之间的同步与互斥	182
5.1.3 线程属性	187
5.1.4 多线程实验	191
5.2 本章小结	196
5.3 本章习题	196
第6章 嵌入式 Linux 网络编程	197
6.1 TCP/IP 协议概述	198
6.1.1 TCP/IP 的分层模型	198
6.1.2 TCP/IP 分层模型的特点	199
6.1.3 TCP/IP 核心协议	200
6.2 网络编程基础	203
6.2.1 套接字 (socket) 概述	203
6.2.2 地址及顺序处理	204
6.2.3 套接字编程	210

6.2.4 编程实例	215
6.3 网络高级编程	218
6.4 广播与组播	225
6.5 实验内容——NTP 协议的客户端实现	231
6.5.1 NTP 协议的客户端实现	231
6.5.2 ARP 断网攻击实验	238
6.6 本章小结	241
6.7 本章习题	242
第 7 章 嵌入式 Linux 设备驱动编程	243
7.1 设备驱动编程基础	244
7.1.1 Linux 设备驱动概述	244
7.1.2 Linux 内核模块编程	246
7.2 字符设备驱动编程	256
7.2.1 字符设备驱动编写流程	256
7.2.2 重要的数据结构	256
7.2.3 设备驱动程序主要组成	257
7.3 GPIO 驱动程序实例	264
7.3.1 GPIO 工作原理	264
7.3.2 GPIO 驱动程序	266
7.4 按键驱动程序实例	275
7.4.1 Linux 设备树	275
7.4.2 中断编程	278
7.4.3 按键工作原理	279
7.4.4 按键驱动程序	280
7.4.5 按键驱动测试	281
7.5 本章小结	282
7.6 本章习题	282
第 8 章 Android 应用编程	283
8.1 Android 发展简史	284
8.2 Android 应用开发环境	286
8.2.1 Android 体系架构简介	286



8.2.2	搭建 Android 应用开发环境	287
8.2.3	编写第一个 Android 应用程序	291
8.3	Android 的四大组件和 Intent	295
8.3.1	Activity	295
8.3.2	Service	300
8.3.3	BroadcastReceiver	301
8.3.4	ContentProvider	302
8.3.5	Intent	303
8.4	表明 Android 常用图形界面组件	305
8.4.1	Android 中的布局管理器	305
8.4.2	TextView	308
8.4.3	EditText	309
8.4.4	Button	311
8.4.5	ImageButton	313
8.4.6	ImageView	313
8.4.7	ProgressBar/SeekBar	313
8.5	本章小结	317
8.6	本章习题	317
第 9 章	Android 播放器项目设计	318
9.1	简易音乐播放器应用的需求	319
9.2	界面设计	321
9.3	播放器控制和播放功能的实现	323
9.4	项目运行	328
9.5	本章小结	340
9.6	本章习题	340

第 1 章

搭建嵌入式 Linux 开发环境

本章讲解嵌入式应用开发的第一步，主要学习如何搭建嵌入式 Linux 开发的环境。从交叉编译环境等嵌入式开发环境的搭建开始，介绍了 Bootloader 的概念及 U-Boot 的编译和移植的方法；然后介绍了 Linux 内核的相关知识，讲解了内核编译和移植的方法。最后介绍了嵌入式 Linux 文件系统的构建。

本章主要内容

- ☐ 构建嵌入式 Linux 交叉开发环境。
- ☐ Bootloader。
- ☐ Linux 内核与移植。
- ☐ 嵌入式文件系统构建。



1.1

构建嵌入式 Linux 交叉开发环境



构建开发环境是任何开发工作的基础，对于软、硬件非常丰富的嵌入式系统来说，构建高效、稳定的环境是能否开展工作的重要因素之一。本节将介绍如何构建一套嵌入式 Linux 开发环境。在构建开发环境之前，有必要了解嵌入式 Linux 开发流程。因为嵌入式 Linux 开发往往会涉及多个层面，这与桌面开发有很大不同。构建一个 Linux 系统，须仔细考虑下面几点：

- 选择嵌入式 Linux 发行版。商业的 Linux 发行版是用于产品开发维护的，经过严格的测试验证，并且可以得到厂家的技术支持。它为开发者提供了可靠的软件和完整的开发工具包。
- 熟悉开发环境和工具。交叉开发环境是嵌入式 Linux 开发的基本模型。Linux 环境配置、GNU 工具链、测试工具甚至集成开发环境都是开发嵌入式 Linux 的利器。
- 熟悉 Linux 内核。因为嵌入式 Linux 开发一般需要重新定制 Linux 内核，所以熟悉内核配置、编译和移植很重要。
- 熟悉目标板引导方式。开发板的 Bootloader 负责硬件平台最基本的初始化，并且具备引导 Linux 内核启动的功能。由于硬件平台是专门定制的，一般需要修改编译 Bootloader。
- 熟悉 Linux 根文件系统。高级一点的操作系统一般都有文件系统的支持，Linux 同样离不开文件系统。系统启动必需的程序和文件都必须放在根文件系统中。Linux 系统支持的文件系统种类非常多，我们可以通过 Linux 内核命令行参数指定要挂接的根文件系统。
- 理解 Linux 内存模型。Linux 是保护模式的操作系统。内核和应用程序分别运行在完全分离的虚拟地址空间，物理地址必须映像到虚拟地址才能访问。
- 理解 Linux 调度机制与进程和线程编程。Linux 调度机制影响到任务的实时性，理解调度机制可以更好地运用任务优先级。此外，进程和线程编程是应用程序开发所必需的。

1.1.1 搭建嵌入式交叉编译环境

搭建交叉编译环境是嵌入式开发的第一步，也是关键的一步。不同的体系结构、不同的操作内容甚至是不同版本的内核，都会用到不同的交叉编译器。选择交叉编译器非常重要，有些交叉编译器经常会有部分的 Bug，这些都会导致最后的代码无法正常运行。

交叉编译器完整的安装一般涉及多个软件的安装（读者可以从 <ftp://gcc.gnu.org/pub/>

下载), 包括 binutils、gcc、glibc、glibc-linuxthreads 等软件。其中, binutils 主要用于生成一些辅助工具, 如 readelf、objcopy、objdump、as、ld 等; gcc 是用来生成交叉编译器的, 主要生成 arm-linux-gcc 交叉编译工具 (应该说, 生成此工具后已经搭建起交叉编译环境, 可以编译 Linux 内核了, 但由于没有提供标准用户函数库, 用户程序还无法编译); glibc 主要是提供用户程序所使用的一些基本的函数库, glibc-linuxthreads 是线程相关函数库。这样, 交叉编译环境就完全搭建起来了。

上面所述的搭建交叉编译环境比较复杂, 很多步骤都涉及对硬件平台的选择。因此, 现在嵌入式平台社区或厂商一般会提供在各种平台上测试通过的交叉编译器, 或把以上安装步骤全部写入脚本文件或者以发行包的形式提供, 这样就大大方便了用户的使用。

在本书中采用广泛使用的 cross-4.6.4 交叉编译器工具链, 其使用方法非常简单。

```
$ mkdir -p /usr/local/arm /* 交叉编译器安装目录*/
$ cp cross-4.6.4.tar.bz2 /usr/local/arm
$ cd /usr/local/arm
$ tar xvf cross-4.6.4.tar.bz2
```

此时在/usr/local/arm/ cross-4.6.4/bin/下已经出现了很多交叉编译工具, 显示如下:

```
arm-arm1176jzfssf-linux-gnueabi-addr2line
arm-arm1176jzfssf-linux-gnueabi-ar
arm-arm1176jzfssf-linux-gnueabi-as
arm-arm1176jzfssf-linux-gnueabi-c++
arm-arm1176jzfssf-linux-gnueabi-cc
arm-arm1176jzfssf-linux-gnueabi-c++filt
arm-arm1176jzfssf-linux-gnueabi-cpp
arm-arm1176jzfssf-linux-gnueabi-ct-ng.config
arm-arm1176jzfssf-linux-gnueabi-elfedit
arm-arm1176jzfssf-linux-gnueabi-g++
arm-arm1176jzfssf-linux-gnueabi-gcc
arm-arm1176jzfssf-linux-gnueabi-gcc-4.6.4
arm-arm1176jzfssf-linux-gnueabi-gcov
arm-arm1176jzfssf-linux-gnueabi-gprof
arm-arm1176jzfssf-linux-gnueabi-ld
arm-arm1176jzfssf-linux-gnueabi-ld.bfd
arm-arm1176jzfssf-linux-gnueabi-ldd
arm-arm1176jzfssf-linux-gnueabi-nm
arm-arm1176jzfssf-linux-gnueabi-objcopy
arm-arm1176jzfssf-linux-gnueabi-objdump
arm-arm1176jzfssf-linux-gnueabi-populate
arm-arm1176jzfssf-linux-gnueabi-ranlib
arm-arm1176jzfssf-linux-gnueabi-readelf
arm-arm1176jzfssf-linux-gnueabi-size
arm-arm1176jzfssf-linux-gnueabi-strings
arm-arm1176jzfssf-linux-gnueabi-strip
arm-linux-addr2line
arm-linux-ar
arm-linux-as
```




从实践中学嵌入式 Linux 应用程序开发（第2版）

```
arm-linux-c++
arm-linux-cc
arm-linux-c++filt
arm-linux-cpp
arm-linux-ct-ng.config
arm-linux-elfedit
arm-linux-g++
arm-linux-gcc
arm-linux-gcc-4.6.4
arm-linux-gcov
arm-linux-gprof
arm-linux-ld
arm-linux-ld.bfd
arm-linux-ldd
arm-linux-nm
arm-linux-objcopy
arm-linux-objdump
arm-linux-populate
arm-linux-ranlib
arm-linux-readelf
arm-linux-size
arm-linux-strings
arm-linux-strip
arm-none-linux-gnueabi-addr2line
arm-none-linux-gnueabi-ar
arm-none-linux-gnueabi-as
arm-none-linux-gnueabi-c++
arm-none-linux-gnueabi-cc
arm-none-linux-gnueabi-c++filt
arm-none-linux-gnueabi-cpp
arm-none-linux-gnueabi-ct-ng.config
arm-none-linux-gnueabi-elfedit
arm-none-linux-gnueabi-g++
arm-none-linux-gnueabi-gcc
arm-none-linux-gnueabi-gcc-4.6.4
arm-none-linux-gnueabi-gcov
arm-none-linux-gnueabi-gdb
arm-none-linux-gnueabi-gprof
arm-none-linux-gnueabi-ld
arm-none-linux-gnueabi-ld.bfd
arm-none-linux-gnueabi-ldd
arm-none-linux-gnueabi-nm
arm-none-linux-gnueabi-objcopy
arm-none-linux-gnueabi-objdump
arm-none-linux-gnueabi-populate
arm-none-linux-gnueabi-ranlib
arm-none-linux-gnueabi-readelf
arm-none-linux-gnueabi-size
arm-none-linux-gnueabi-strings
arm-none-linux-gnueabi-strip
```