

ARM 嵌入式Linux系统开发丛书



引领ARM嵌入式Linux编程应用开发新潮流!

ARM嵌入式 Linux 应用开发入门

汪明虎 欧文盛 编著

本书内容包括:

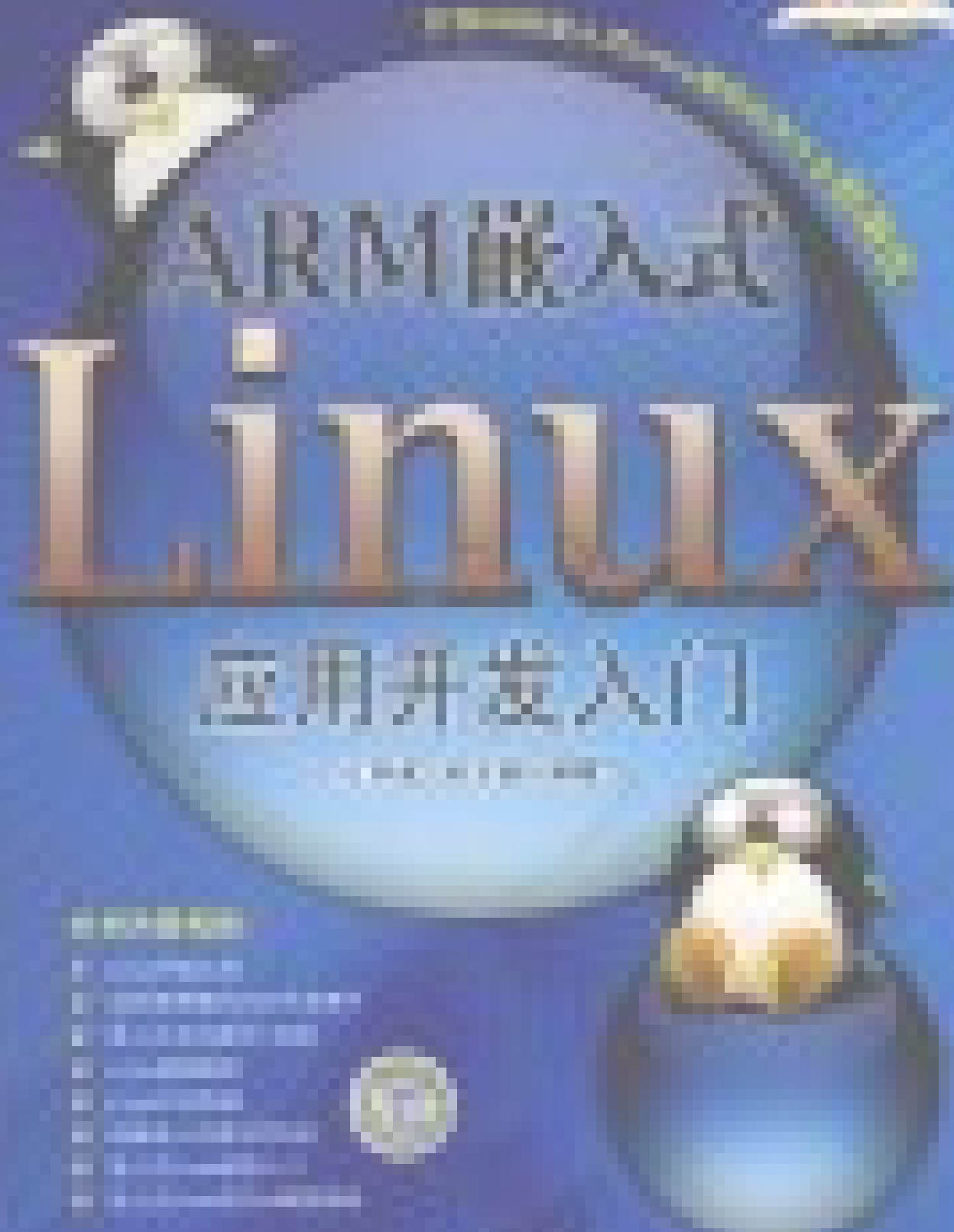
- Linux开发工具
- ARM处理器和ADS开发套件
- 嵌入式交叉编译工具链
- Linux编程基础
- Linux内核移植
- 构建嵌入式根文件系统
- 嵌入式Linux驱动入门
- 嵌入式Linux的GUI编程基础



中国电力出版社
www.infopower.com.cn



清华大学出版社“十二五”国家重点图书出版规划项目



内容简介

本书以 ARM 嵌入式 Linux 应用开发为主线，从 Linux 系统的安装、配置、编译、运行、调试、维护等方面入手，详细介绍了 ARM 嵌入式 Linux 系统的开发流程。本书可作为高等院校计算机专业及相关专业的教材，也可供从事 ARM 嵌入式 Linux 应用开发的工程技术人员参考。



清华大学出版社

ARM 嵌入式



Linux 系统开发丛书

TP332/143

:1

2008

ARM 嵌入式 Linux 应用开发入门

汪明虎 欧文盛 编著



中国电力出版社

www.infopower.com.cn

内 容 提 要

本书是《ARM 嵌入式 Linux 系统开发丛书》之一。

本书针对嵌入式 Linux 开发的特点,结合工程开发实践,使读者能够边学边用,从而快速地掌握相关的知识要点。全书共分为 10 章,首先介绍了嵌入式系统开发的基础知识,包括 Linux 开发工具、ARM 处理器和 ADS 开发套件等。然后重点讲解了嵌入式交叉编译工具链、Linux 编程基础、BootLoader 移植、Linux 内核移植、构建嵌入式根文件系统和嵌入式 Linux 驱动入门,以及嵌入式 Linux 的 GUI 编程基础。

本书可作为相关工程技术人员嵌入式培训教材及参考用书,也可作为高等院校自动化类、电子类、电气类、计算机类专业本科生、研究生学习嵌入式 Linux 开发的辅助教材。

图书在版编目(CIP)数据

ARM 嵌入式 Linux 应用开发入门 / 汪明虎, 欧文盛编著. —北京: 中国电力出版社, 2008
(ARM 嵌入式 Linux 系统开发丛书)
ISBN 978-7-5083-7437-6

I. A… II. ①汪…②欧… III. ①微处理器, ARM - 系统设计②Linux 操作系统 - 系统设计
IV. TP332 TP316.89

中国版本图书馆 CIP 数据核字(2008)第 077262 号

责任编辑: 刘 焱
责任校对: 崔燕菊
责任印制: 郭华清

书 名: ARM 嵌入式 Linux 应用开发入门
编 著: 汪明虎 欧文盛
出版发行: 中国电力出版社

地址: 北京市三里河路 6 号 邮政编码: 100044
电话: (010) 68362602 传真: (010) 68316497

印 刷: 北京同江印刷厂

开本尺寸: 185mm × 260mm 印 张: 15.5 字 数: 325 千字

书 号: ISBN 978-7-5083-7437-6

版 次: 2008 年 7 月北京第 1 版

印 次: 2008 年 7 月第 1 次印刷

印 数: 0001—4000 册

定 价: 28.00 元

敬 告 读 者

本书封面贴有防伪标签,加热后中心图案消失
本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

PREFACE

前

言

随着计算机控制和电子技术的快速发展,以及大规模集成电路的产生而出现的微型计算机,使现代科学研究得到了质的飞跃,特别是嵌入式微控制器技术的出现,给现代工业控制领域带来了一次新的技术革命。目前 ARM 在嵌入式应用领域的霸主地位是不容撼动的,同时 Linux 作为一个备受关注、开放源代码的操作系统,在嵌入式应用领域里也获得了长足的发展。

嵌入式 Linux 属于一个交叉学科,并且也是一个高起点的学科,它涵盖了微电子技术、电子信息技术、计算机软硬件等多项技术领域的综合应用。另外,学习嵌入式 Linux 不仅要求具备相应的嵌入式开发板和软件,还需要有经验的人进行指导。目前国内大部分高校都很难达到这种要求,这也造成了目前国内嵌入式 Linux 开发人才匮乏的局面。而许多人在嵌入式开发的学习上徘徊不前、无从下手,这种局面是急需解决的。

本书针对国内学习嵌入式 Linux 的人员当前的状况,详细讲解了嵌入式 Linux 开发所必需的基础知识,使读者能够通过本书的学习,打开嵌入式学习和应用开发的大门,从而登堂入室,进入嵌入式 Linux 知识的殿堂。

本书第 1~5 章由汪明虎编写,第 6~9 章由欧文盛编写,李亚锋负责审核。

主要内容

本书共由 10 个章节组成,以下是每个章节的内容概要。

- 第 1 章:介绍嵌入式系统概念、特点、发展趋势,嵌入式系统软件开发流程,以及 Linux 常用编程工具,包括 vi、GCC、gdb、makefile 等的常用命令和使用技巧,并通过实例的方式进行讲解,使读者更容易理解和掌握。
- 第 2 章:简要介绍 ARM 处理器的硬件平台及 ADS 开发工具的使用方法。首先对 ARM 处理器进行介绍,然后讲解 ARM9 处理器的特点、工作模式、工作状态。还介绍了本书对相关代码进行调试时使用的开发板的软硬件配置情况。最后介绍了 ADS 开发工具,包括命令行开发工具,以实例的形式讲解了 Code Warrior 开发环境和 AXD 调试器的使用。
- 第 3 章:介绍如何构建嵌入式交叉编译工具链。首先介绍了交叉编译环境及构建交叉编译工具链的方法。然后通过实例来介绍如何用 Crosstool 构建交叉编译工具链,以及如何分步构建嵌入式交叉编译工具链。
- 第 4 章:介绍 Linux 编程基础知识,涉及到 Linux 文件 I/O 操作、进程的产生、退出、竞态条件、控制、进程间的通信方法——管道、消息队列、共享内存、Linux 网络编程等,几乎每部分都配有实例讲解。
- 第 5 章:介绍嵌入式系统 BootLoader——u-boot 的移植。首先介绍了 BootLoader



的种类、工作原理, 然后对 u-boot 的启动代码进行分析, 并对移植步骤加以详细讲解, 对其中涉及到的 ARM 汇编进行同步介绍, 方便不熟悉汇编的读者进行学习。

- ❑ 第 6 章: 介绍了 Linux 内核的移植, 主要包括 NANDFlash 分区的设置、内核 makefile 的修改、配置内核选项、编译下载内核, 内容具体翔实, 有利于读者动手操作。
- ❑ 第 7 章: 介绍什么是根文件系统及其分类、怎样选择适合的根文件系统、如何定制根文件系统、如何建立 Cramfs 根文件系统映像, 并配有实用的例子。通过本章的学习, 读者将能够自己创建根文件系统。
- ❑ 第 8 章: 介绍嵌入式 Linux 驱动开发入门知识。首先介绍嵌入式 Linux 驱动开发的基本知识, 然后通过实例来说明 Linux 内核驱动模块程序的一些基本特点, 最后讲解如何编写字符设备驱动。
- ❑ 第 9 章: 介绍嵌入式 Linux 系统下的 GUI 编程, 主要内容包括嵌入式 Linux 下 GUI 系统概述、Qt/ia Core 编程基础及移植、MiniGUI 编程基础及移植等。

本书特点

Linux 是一个成熟而稳定的开放源代码的操作系统, 将 Linux 移植到嵌入式设备中具有众多的优点。本书是嵌入式 Linux 开发的入门读物, 主要介绍嵌入式 Linux 开发所涉及到的基础知识。由于嵌入式 Linux 开发起点比较高, 入门比较难, 许多读者的主要问题是找不到切入点。本书就是针对嵌入式 Linux 开发的特点, 将嵌入式开发所需用到的基础知识, 包括 Linux 开发工具、ARM 处理器、ADS 开发套件加以详细介绍。然后介绍学习嵌入式开发所必备的本领, 包括编译嵌入式交叉编译工具链、Linux 编程基础、Linux BootLoader 移植、Linux 内核移植、构建嵌入式根文件系统、嵌入式 Linux 驱动入门及嵌入式 Linux 的 GUI 编程基础。读者掌握了这些知识, 就基本达到嵌入式 Linux 学习的入门级别, 为进一步从事嵌入式开发工作打下坚实的基础。总之, 本书对那些想从事嵌入式开发或刚刚从事嵌入式开发的读者来说无疑是一本难得的参考书。

读者对象

本书可作为高等院校电子信息、自动化和计算机等专业的本科生、研究生学习嵌入式 Linux 开发入门的参考书或自学教材, 也可供广大希望转入嵌入式领域的科研和工程技术人员参考使用, 还可作为嵌入式培训班的教材和辅导材料。

致谢

本书在编写过程中, 得到了许多支持和帮助, 首先感谢我妻子章丽平的默默支持, 同时感谢同学李亚锋、欧文盛的宝贵建议和帮助, 另外本书还得到了汪志国、江灏、邓文全、汪明发、梁娟娟、张广忠、李俊坤、汪明咤、李艳、王福东、白俊、蔡传玲、翟亚东、白海、张静、高兰涛的大力支持和帮助, 在此一并致谢。在编写本书的过程中我感受到同学和朋友之间深厚的情谊, 正是大家的互相支持、互相提携, 才使本书得以顺利完成。

鉴于作者水平有限, 加之时间仓促, 本书一定有不少疏漏与不妥之处, 恳请广大读者批评与指正。

作 者

2008 年 4 月

CONTENTS

目

录

前 言

第 1 章 嵌入式系统概述及 Linux 开发工具	1
1.1 嵌入式系统简介	2
1.1.1 什么是嵌入式系统	2
1.1.2 嵌入式系统基本结构	2
1.1.3 嵌入式系统特点	3
1.2 嵌入式软件的发展	3
1.3 嵌入式系统软件开发流程	4
1.4 使用 Linux 开发工具	4
1.4.1 vi 编辑器	4
1.4.2 GCC 编译器	8
1.4.3 gdb 调试器	13
1.4.4 编写 makefile	16
1.5 小结	20
第 2 章 ARM 硬件平台及 ADS 开发工具	21
2.1 ARM 处理器简介	22
2.2 ARM9 处理器	23
2.2.1 ARM9 处理器的特点	23
2.2.2 ARM9 处理器的工作模式	23
2.2.3 ARM 处理器状态	24
2.3 LJD-2410DVK-I 开发板	25
2.3.1 硬件资源	25
2.3.2 软件资源	27
2.4 ADS 开发工具	27
2.4.1 命令行开发工具	29
2.4.2 Code Warrior 开发环境初步	32



2.4.3	AXD 调试器	42
2.5	小结	52
 第 3 章 构建嵌入式交叉编译工具链		53
3.1	什么是交叉编译环境	54
3.2	构建交叉编译工具链的方法	54
3.3	用 Crosstool 构建交叉编译工具链	54
3.3.1	Crosstool 简介	54
3.3.2	下载软件包	55
3.3.3	配置脚本文件	55
3.3.4	生成交叉编译工具链	57
3.3.5	测试生成的编译工具	57
3.4	分步构建嵌入式交叉编译工具链	58
3.4.1	创建目录	58
3.4.2	建立环境变量	59
3.4.3	编译 binutils	59
3.4.4	配置内核头文件	61
3.4.5	编译 GCC (阶段 1)	61
3.4.6	安装 glibc	62
3.4.7	编译 GCC (阶段 2)	63
3.4.8	测试交叉编译工具链	63
3.5	小结	63
 第 4 章 嵌入式系统 Linux 编程基础		65
4.1	文件 I/O 编程	66
4.1.1	文件描述符	66
4.1.2	open 函数	66
4.1.3	close 函数	67
4.1.4	lseek 函数	67
4.1.5	read 函数	67
4.1.6	write 函数	68
4.1.7	dup 和 dup2 函数	69
4.1.8	fcntl 函数	69
4.2	进程控制开发	70
4.2.1	fork 和 vfork 函数	71

4.2.2	exit 函数和_exit 函数	72
4.2.3	wait 和 waitpid 函数	73
4.2.4	竞态条件	74
4.2.5	exec 函数	75
4.3	进程间通信	77
4.3.1	管道 pipe	77
4.3.2	有名管道 FIFO	79
4.3.3	消息队列	83
4.3.4	信号量 (semaphore)	86
4.3.5	共享内存	89
4.4	Linux 网络编程	92
4.4.1	TCP/IP 协议简介	92
4.4.2	常用网络编程函数	93
4.4.3	套接口使用实例	97
4.5	小结	98
第 5 章 u-boot 代码分析及移植		99
5.1	BootLoader 简介	100
5.1.1	BootLoader 分类	100
5.1.2	BootLoader 操作模式	100
5.1.3	BootLoader 工作流程	101
5.2	u-boot 代码分析	101
5.2.1	u-boot 主要目录结构	101
5.2.2	u-boot 代码分析	102
5.3	移植 u-boot	114
5.3.1	获取源码	114
5.3.2	创建交叉编译环境	114
5.3.3	和移植相关的文件夹	114
5.3.4	移植步骤	115
5.3.5	修改相关文件	117
5.4	烧写 BootLoader 到开发板	131
5.5	小结	132
第 6 章 嵌入式 Linux 内核移植		133
6.1	NAND Flash 分区	134



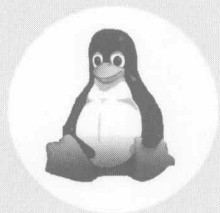
6.2	配置内核.....	137
6.2.1	修改 makefile 文件.....	137
6.2.2	添加 devfs 配置.....	137
6.2.3	配置内核选项.....	138
6.3	内核编译.....	145
6.3.1	编译 zImage.....	145
6.3.2	编译和安装模块.....	146
6.3.3	如何处理编译出错.....	146
6.4	下载内核到开发板.....	148
6.5	小结.....	149
第 7 章 构建根文件系统.....		151
7.1	根文件系统概述.....	152
7.1.1	什么是根文件系统.....	152
7.1.2	根文件系统的基本目录结构.....	153
7.2	定制根文件系统.....	154
7.2.1	创建空的目录树.....	154
7.2.2	移植 BusyBox.....	154
7.2.3	选择必要的动态共享库.....	160
7.2.4	初始化脚本.....	162
7.2.5	模块和设备文件.....	163
7.3	选择合适的 Flash 文件系统.....	164
7.3.1	嵌入式 Linux 文件系统.....	164
7.3.2	存储设备 NOR/NAND Flash.....	166
7.3.3	基于 Flash 的文件系统类型.....	167
7.4	建立 Cramfs 根文件系统映像.....	169
7.4.1	Cramfs 工具包的使用.....	169
7.4.2	构建 Cramfs 根文件系统.....	170
7.5	小结.....	171
第 8 章 嵌入式 Linux 驱动开发入门.....		173
8.1	Linux 驱动程序概述.....	174
8.2	Linux 驱动设计基础知识.....	175
8.2.1	内核模块.....	176
8.2.2	设备驱动基本类型.....	176

8.2.3	主、次设备号	177
8.2.4	字符设备相关的数据结构	179
8.3	Helloworld 模块实例	186
8.3.1	源代码及分析	186
8.3.2	编写 makefile	188
8.3.3	测试 Helloworld 模块	190
8.4	字符设备驱动实例	191
8.4.1	源代码及分析	191
8.4.2	测试代码及分析	196
8.4.3	编译和测试	197
8.5	小结	198
第 9 章 嵌入式 Linux 系统 GUI 编程入门		199
9.1	嵌入式 Linux 下 GUI 系统概述	200
9.1.1	传统的 Linux 桌面 GUI 系统	200
9.1.2	嵌入式 Linux 下的 GUI 系统	202
9.2	Qt/ia Core 编程基础及移植	203
9.2.1	Qt/X11、Qt/Embedded 和 Qt/ia Core	204
9.2.2	Qt/ia Core 的安装	205
9.2.3	Frame Buffer 和 qvfb	207
9.2.4	编写简单的 Qt/ia Core 程序	213
9.2.5	移植 Qt/ia Core 到开发板	215
9.2.6	Qt 的特性简介	217
9.3	MiniGUI 编程基础及移植	225
9.3.1	MiniGUI 的下载与授权问题	226
9.3.2	安装 MiniGUI	226
9.3.3	编写及编译运行 MiniGUI 程序	229
9.3.4	移植 MiniGUI 到开发板	232
9.4	小结	235
参考文献		236

第 1 章

嵌入式系统概述及 Linux 开发工具

- ▶ 1.1 嵌入式系统简介
- ▶ 1.2 嵌入式软件的发展
- ▶ 1.3 嵌入式系统软件开发流程
- ▶ 1.4 使用 Linux 开发工具
- ▶ 1.5 小结





1.1

嵌入式系统简介

本节主要针对嵌入式系统的定义、嵌入式系统的结构以及嵌入式系统的特点进行介绍。首先来介绍嵌入式系统的定义。

1.1.1 什么是嵌入式系统

嵌入式系统被定义为：以应用为中心，以计算机技术为基础，软件硬件可裁剪、适应应用系统，对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。嵌入式系统是为特定目的而构建的一套类计算机设备。该设备的主要特点是适应非常严格的特殊要求，如体积、功耗、可靠性等，而且嵌入式系统是以应用为中心的系统，应用决定了其软件、硬件的组成，具有软件代码小、高度自动化、响应速度快等特点，特别适合于要求实时和多任务的体系。

嵌入式系统几乎包括了生活中的所有电器设备，如掌上 PDA、移动计算设备、电视机顶盒、手机、数字电视、多媒体、汽车、微波炉、数码相机、智能家居系统、电梯、空调、安全系统、自动售货机、工业自动化仪表与医疗仪器等。嵌入式设备在民用品和军事装备上得到了广泛应用。

1.1.2 嵌入式系统基本结构

嵌入式系统作为一类比较特殊的计算机系统，自下向上包含有 4 个部分，如图 1.1 所示。

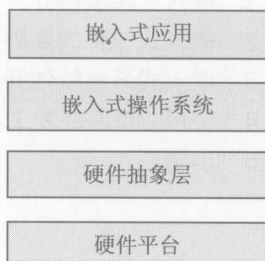


图 1.1 嵌入式系统基本结构

(1) 硬件平台是整个嵌入式操作系统和应用程序运行的硬件平台，不同的应用通常有不同的硬件环境。硬件平台的多样性是嵌入式系统的一个主要特点。

(2) 硬件抽象层，也叫板级支持包，是一个介于硬件与软件之间的中间层次。硬件抽象层通过特定的上层接口与操作系统进行交互，向下与硬件进行交互。

(3) 嵌入式操作系统具有实时性、可靠性、强壮性、容错性、可裁剪可扩展性、可移植性、基于强占式调度策略的微内核系统,是为了简化不同平台的程序开发、移植、功能集中、精炼地对有限的资源进行充分利用与管理的操作系统。

(4) 嵌入式应用程序是运行于操作系统之上,利用操作系统提供的机制完成特定功能的嵌入式应用。

1.1.3 嵌入式系统特点

嵌入式系统与通用系统的最大的区别是,嵌入式系统是一种竞争激烈、不可垄断的工业。各种体系结构和操作系统相互竞争、相互促进,没有哪一个系统能够一统天下。各种不同的需求要求不同的系统,所以新系统层出不穷,也就形不成垄断局面。所以嵌入式系统各个领域的发展空间非常大。

嵌入式系统主要是面向应用的。不同的应用要求不同的硬件和软件来达到最低功耗,体积和容量也都要最小,从而实现用户的需求。一般软件都要求固化在存储器芯片或嵌入式微控制器本身中。软件代码要求质量高、可靠性高,操作系统也要求有很高的实时性,而且嵌入式系统的开发需要特定开发环境和相关调试工具。

1.2

嵌入式软件的发展

进入 20 世纪 90 年代以来,以计算机技术、通信技术和软件技术为核心的信息技术取得了更加迅猛的发展,各种装备与设备上嵌入式计算与系统的广泛应用大大地推动了行业的发展。几乎所有的嵌入式产品中都需要嵌入式软件来提供灵活多样、针对应用的功能。由于嵌入式系统应用广泛,嵌入式软件在整个软件产业中占据了重要地位,并受到世界各国的广泛关注。

嵌入式软件产业发展迅猛,已成为软件体系的重要组成部分。嵌入式系统产品正不断渗透到各个行业,嵌入式软件作为包含在硬件产品中的特殊软件形态,其产业增幅不断加大,而且在整个软件产业的比重日趋提高。

汽车、通信、信息、电器、医疗、军事等行业的巨大智能化装备需求推动了嵌入式软件及系统的发展。同传统的通用计算机系统不同,嵌入式系统面向特定应用领域,根据应用需求定制开发,并随着智能化产品的普遍需求渗透到各行各业。随着硬件技术的不断革新,硬件平台的处理能力不断增强,硬件成本不断下降,嵌入式软件已成为产品的数字化改造、智能化增值的关键性、带动性技术。

SOC(片上系统)技术是微电子技术发展的一个新的里程碑,并已成为当今超大规模 IC 的发展趋势,为 IC 产业提供前所未有的广阔市场和难得的发展机遇。迅猛发展的 SOC 工业再次推进了嵌入式软件与硬件系统进一步融合,而嵌入式软件是其灵魂与核心。



大量的嵌入式设备急需网络连接来提升其服务能力和应用价值。这将催生嵌入式技术的发展和更新。

随着嵌入式微处理器的飞速发展，从 8 位到 16 位、32 位、64 位，推动了嵌入式系统软件向运行速度更快、支持功能更强、应用开发更便捷的方向不断发展。

总之，嵌入式系统具有广阔的发展前景。同样，嵌入式软件也在不断地发展进步，应用越来越广泛，而且向着集成化、自动化、高可靠性方向发展。

1.3

嵌入式系统软件开发流程

嵌入式系统开发主要分为应用开发、驱动程序开发、嵌入式操作系统移植和开发、BootLoader 移植和开发。要想从事嵌入式开发，必须要建立好嵌入式编程环境。就 ARM 系统上进行 Linux 开发而言，具体包括如下内容：

- (1) 选择符合自己学习要求的目标硬件系统。例如选择 ARM 微处理器 Flash 等。
- (2) 建立交叉编译工具。嵌入式系统编译工具一般需要自己建立。
- (3) 开发或移植 BootLoader：这是系统的引导程序。
- (4) 移植 Linux 内核。
- (5) 开发根文件系统。
- (6) 开发相关硬件的驱动程序。
- (7) 开发上层的应用程序。

1.4

使用 Linux 开发工具

学习嵌入式 Linux，离不开 Linux 开发工具。本节将介绍常见的 Linux 开发工具的使用方法，使读者掌握这些工具的基本知识。

Linux 编程环境中虽然也提供类似 Windows 中的所有功能都包含的 IDE，但 Linux 提供的单独的编程工具功能非常强大，而且用得非常多。我们可以自己处理一些事情而不是依靠 IDE，来使我们更清楚地了解系统是如何工作的。下面介绍 Linux 环境中的编程需要用到的一些基本工具。本节主要介绍 vi 编辑器、GCC 编译器、gdb 调试器及如何编写 makefile。

1.4.1 vi 编辑器

在 Linux 中，可以在图形界面下进行文本的编辑，就像在 Windows 下的文本编辑器中

进行编辑一样,非常方便。Linux 图形界面的文本编辑器有很多,如 gedit、kwrite 等。但在 Linux 中,经常会在控制台下进行编程,这时就要用到 Linux 文本模式编辑器了。文本模式下的编辑器有 vi、vim (vi 的增强版本)、Emacs 和 nano 等。vi 和 vim 是 Linux 中最常用的编辑器。这里就 vi 编辑器进行具体介绍,使读者能熟悉其基本功能。Emacs 也是一款功能强大的编辑器,如果读者对 Emacs 感兴趣,可以参考相关资料进行学习。

vi 是 Visual interface 的简称,它在 Linux 中的地位就像 Edit 程序在 DOS 中一样。它可以执行输出、删除、查找、替换、块操作等众多文本操作,而且用户可以根据自己的需要对其进行定制,这是其他编辑程序所不能的。

vi 没有菜单,只有命令,且命令繁多。vi 有两种基本工作模式:命令行模式和文本输入模式。另外,vim 中还有一种图形模式,但只有 vim 才支持,这里不作介绍。

首先介绍 vi 的基本操作方式。

► 1. 启动 vi

```
#vi filename
```

filename 即要创建的文件名。启动 vi 编辑器可以直接跟文件名,也可以直接输入 vi,不跟文件名来启动,待文件编辑完成后保存时,系统会提示输入文件名。

► 2. vi 的两种模式

(1) Command (命令) 模式,用于输入命令。

(2) Insert (插入) 模式,用于编辑文本。

Command 模式是 vi 的默认模式,即打开 vi 时即处于此状态。在命令模式下,用户可以输入各种命令,如保存文件等。在文本模式下,可以对文本进行编辑,如插入字符、修改字符等。如果想从命令模式切换到文本模式,按 i 键即可。如果想从插入模式切换到 Command 模式,按 Esc 键即可切换,多次按 Esc 键等同于按一次。

► 3. 文件的保存和退出

如果想保存或退出 vi,就需要进入命令模式。如在插入模式下,则需要先按 Esc 键切换到命令模式。然后输入冒号,这时光标会出现在屏幕的最下方,并等待用户输入命令。命令包括:

- w: 保存不退出。
 - w filename: 另存为 filename。
 - wq!: 保存退出。
 - wq! Filename: 以 filename 为文件名保存后退出。
 - q!: 不保存退出。
- 输入命令后回车即可。



注意

上述命令如 w 后面的冒号不是命令的组成部分,而是直接输入 w 后回车即可达到保存不退出的效果。后述命令类似。



► 4. 光标移动

在文本模式下可以移动光标来进行编辑，即使用上下左右方向键。在命令模式下也可以移动光标，这可以加快编辑速度，若能熟练操作，则非常方便。具体命令如下：

- j: 向下移动一行。
- k: 向上移动一行。
- h: 向左移动一个字符。
- l: 向右移动一个字符。
- ctrl+b: 向上移动一屏。
- ctrl+f: 向下移动一屏。
- 向上箭头: 向上移动。
- 向下箭头: 向下移动。
- 向左箭头: 向左移动。
- 向右箭头: 向右移动。

对于 j、k、l 和 h 键，可以在这些动作命令的前面加上数字，例如 4j 表示向下移动 4 行。

► 5. 文本的插入

命令模式下的文本插入操作（注意大小写）。

- i: 在光标之前插入。
- a: 在光标之后插入。
- I: 在光标所在行的行首插入。
- A: 在光标所在行的行末插入。
- o: 在光标所在行的上面插入一行。
- O: 在光标所在行的下面插入一行。

► 6. 删除操作

命令模式下的删除操作：

- s: 删除光标后的一个字符，然后进入插入模式。
 - S: 删除光标所在的行，然后进入插入模式。
 - Nx: 删除光标后的几个字符，N 表示数字，如 3x 表示删除 3 个字符。
 - dw: 删除光标后的一个单词。
 - Ndw: 删除光标后的几个单词，N 用数字表示，如 3dw 表示删除三个单词。
 - dd: 删除一行。
 - Ndd: 删除多个行，N 代表数字，例如 3dd 表示删除光标所在行及光标下的两行。
 - d\$或 D: 删除光标到行尾的内容。
 - do: 删除光标到行首的内容。
 - J: 清除光标所处的行与下一行之间的空格，把光标所在行和下一行接在一起。
- vi 中的删除同时也表示复制。下面介绍的复制就是在此基础之上进行的操作。