

嵌入式Linux编程 与实践教程

王粉花 主编
李 擎 王尚君 栗 辉 编著



科学出版社

嵌入式 Linux 编程与实践教程

王粉花 主编

李 擎 王尚君 栗 辉 编著

科学出版社

北京

内 容 简 介

本书是一部讲述嵌入式 Linux 编程技术并指导实践的教材。全书结合大量应用实例,详细介绍嵌入式 Linux 系统内核、系统管理、编程基础、基于 C 语言的应用编程技术及嵌入式 Linux 系统的设计开发方法。

全书共 10 章,内容包括嵌入式系统概述、嵌入式 Linux 操作系统基础、嵌入式 Linux 编程基础、嵌入式 Linux 文件编程、嵌入式 Linux 进程控制、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 系统构建、嵌入式 Linux 数据采集系统开发等。其中第 9 章系统介绍开发嵌入式 Linux 系统的方法,包括硬件电路板的设计、交叉编译环境的构建、Bootloader 的移植、嵌入式 Linux 内核的定制与编译等全过程。第 10 章以嵌入式 Linux 数据采集系统为例,详细介绍嵌入式 Linux 应用系统的设计开发过程。

本书可作为高等院校控制类和信息类相关专业嵌入式 Linux 程序设计课程的教材,也可供嵌入式 Linux 技术爱好者和从事嵌入式 Linux 系统设计的开发人员参考使用。

图书在版编目(CIP)数据

嵌入式 Linux 编程与实践教程 / 王粉花主编; 李擎, 王尚君, 栗辉编著.
—北京: 科学出版社, 2016.3
ISBN 978-7-03-047920-4

I. ①嵌… II. ①王… ②李… ③王… ④栗… 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(2016)第 059200 号

责任编辑: 潘斯斯 张丽花 / 责任校对: 郭瑞芝

责任印制: 霍 兵 / 封面设计: 迷底书装

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

文林印务有限公司 印刷

科学出版社发行 各地新华书店经销

*

2016 年 3 月第 一 版 开本: 787×1092 1/16

2016 年 3 月第一次印刷 印张: 18 1/2

字数: 439 000

定价: 48.00 元

(如有印装质量问题, 我社负责调换)

前 言

从 20 世纪 70 年代单片机推出开始,嵌入式技术走过了 40 多年的独立发展道路。随着我国城市信息化和行业信息化的持续深入,嵌入式技术已成为信息产业中发展最快、应用最广的计算机技术之一,并被广泛应用于网络通信、消费电子、医疗电子、工业控制和智能交通等领域,可以说嵌入式技术无处不在。与此同时,嵌入式行业以其应用领域广、人才需求大、行业前景好等诸多优势,成为当前最热门的行业之一。相比之下,由于嵌入式领域入门门槛较高、技术较新,国内专业的嵌入式人才比较紧缺。因此,近几年来,各高校纷纷开设嵌入式系统相关课程。

嵌入式操作系统是指用于嵌入式系统的操作系统,是一种用途广泛的系统软件,负责嵌入式系统的全部软硬件资源的分配、任务调度,控制、协调并发活动,能够通过装载某些模块来达到系统所要求的功能。由于 Linux 具有开放源代码、内核小、效率高、适用于多种 CPU 和多种硬件平台等优势,所以 Linux 特别适合作为嵌入式操作系统。

本书结合大量应用实例,详细介绍嵌入式 Linux 系统内核、系统管理、编程基础、基于 C 语言的应用编程技术及嵌入式 Linux 系统的设计开发方法,可供智能科学与技术、自动化、测控技术与仪器、计算机科学与技术、电子科学与技术、通信工程、信息安全、物联网工程等相关专业高年级本科生学习和研究生选用。本书的案例、习题与实践可供教师课堂教学或学生课后自学使用。

本书总体编写思路如下。

第一部分即第 1 章,整体讲述嵌入式系统的定义与特点,分析其应用领域、发展历程与发展趋势,剖析嵌入式系统的组成结构,重点阐述各种主流嵌入式操作系统的发展历程、应用现状及特点。

第二部分共两章,详细介绍嵌入式 Linux 内核结构、系统管理及编程基础。其中第 2 章分析嵌入式 Linux 内核结构和文件结构,结合操作实例详细介绍嵌入式 Linux 系统配置过程和基本操作命令的使用方法;第 3 章论述嵌入式 Linux 编程基础,重点介绍 GCC 编译器、GDB 程序调试器及 Makefile 工程管理,为嵌入式 Linux 程序设计打下基础。

第三部分共五章,详细论述嵌入式 Linux 基于 C 语言的编程技术。其中第 4 章为嵌入式 Linux 文件编程,首先介绍虚拟文件系统、Linux 文件类型、文件系统组成及文件描述符,然后详细介绍基于 Linux 操作系统的基本文件 I/O 操作和基于 C 语言库函数的标准 I/O 操作的编程方法,包括文件的打开、关闭、读写及定位等编程技术,最后介绍 Linux 时间编程,包括时间的获取、转换、显示及延时等编程技术;第 5 章为嵌入式 Linux 进程控制,首先介绍进程控制理论基础,包括进程定义、进程特点、进程状态、进程 ID、进程互斥、临界资源与临界区、进程同步、进程调度及调度算法、死锁等概念和方法,然后详细介绍进程控制编程技术;第 6 章为嵌入式 Linux 进程间通信,在介绍进程通信目的、发展历程及分类的基础上,详细介绍管道通信、信号通信、共享内存通信、消息队列通信及信号量通信的编程技术;第 7 章为嵌入式 Linux 多线程编程,详细介绍线程设计、线程属性及线程数据处理的编程方法;第 8 章为嵌入式 Linux 网络编程,首先介绍网络模型、TCP/IP 协议族、套接字、网络地址、

字节序等相关概念，然后详细介绍基于套接字的编程方法，最后重点介绍基于 TCP 的网络程序设计技术和基于 UDP 的网络程序设计技术。

第四部分共两章，详细介绍在 ARM9 2440 嵌入式开发板上搭建嵌入式 Linux 系统，并设计开发一个嵌入式 Linux 应用系统的全过程。其中第 9 章为嵌入式 Linux 系统构建，讲述 ARM9 2440 目标板硬件构建和嵌入式系统开发环境搭建方法，包括目标板硬件电路设计、宿主开发环境搭建和基础软件移植等技术；第 10 章以一个嵌入式 Linux 数据采集系统为例，讲述嵌入式 Linux 应用系统的开发过程。

本书在仔细梳理嵌入式 Linux 系统所涵盖主要编程技术的基础上，理出一条学习编程的循序渐进的路线，并详细阐述了其中的关键技术。作为教材，本书具有以下特色。

(1) 知识点由浅入深，内容融会贯通。首先介绍嵌入式系统相关概念和主流嵌入式操作系统，剖析嵌入式 Linux 内核结构，讲述嵌入式 Linux 系统管理方法及编程基础，可以引导读者学会 Linux 系统配置、基本操作命令、GCC 编译器和 GDB 调试器应用及 Makefile 工程管理等技术。在此基础上，论述嵌入式 Linux 基于 C 语言的编程技术，涉及文件编程、进程控制、进程间通信、多线程编程及网络编程等技术，可以使读者掌握 Linux 平台上 C 语言编程的核心知识和技术；最后部分可使读者学会在 ARM9 2440 嵌入式开发板上搭建嵌入式 Linux 系统的方法，进一步开发出一个嵌入式 Linux 应用系统。知识点分布既有连续性，又有一定的跨越性，可满足不同层次教学的需求。

(2) 案例丰富，面向实际应用。针对每一个知识点，本书都设计了相应的编程案例，便于读者学以致用。

(3) 实践指导贯穿全书。本书既可以作为理论教学用教材，也可以作为实践教学用教材。

(4) 例题、习题齐全，方便教学。在总结作者多年教学经验的基础上，本书编排了丰富的例题和习题，非常适合教学和自学。

本书由王粉花主编，负责制定本书编写大纲、指导文字写作和组织工作，栗辉负责全书的统稿工作。王粉花编写了第 1~8 章；李擎编写了第 9 章；王尚君和栗辉编写了第 10 章。

本教材的编写和出版得到了北京科技大学“十二五”教材建设经费的资助。由于作者水平有限，书中难免存在不足之处，敬请广大读者批评指正。作者的电子信箱是：wangfenhua@ustb.edu.cn。

王粉花

2015 年 12 月于北京科技大学

目 录

前言

| | |
|-------------------------------------|----|
| 第 1 章 绪论 | 1 |
| 1.1 嵌入式系统概述 | 1 |
| 1.1.1 嵌入式系统定义与特点 | 1 |
| 1.1.2 嵌入式系统应用领域 | 2 |
| 1.1.3 嵌入式系统发展历程与发展趋势 | 4 |
| 1.1.4 嵌入式系统组成 | 7 |
| 1.2 嵌入式操作系统 | 12 |
| 1.2.1 嵌入式操作系统简介 | 12 |
| 1.2.2 嵌入式 Linux | 12 |
| 1.2.3 Windows Embedded | 20 |
| 1.2.4 VxWorks 操作系统 | 24 |
| 1.2.5 Android 系统 | 25 |
| 1.2.6 iOS 系统 | 28 |
| 本章小结 | 29 |
| 习题与实践 | 29 |
| 第 2 章 嵌入式 Linux 操作系统基础 | 30 |
| 2.1 嵌入式 Linux 操作系统内核结构 | 30 |
| 2.1.1 Linux 操作系统内核结构 | 30 |
| 2.1.2 Linux 操作系统文件结构 | 32 |
| 2.2 嵌入式 Linux 系统管理 | 34 |
| 2.2.1 嵌入式 Linux 系统配置 | 34 |
| 2.2.2 嵌入式 Linux 的基本操作命令 | 38 |
| 本章小结 | 72 |
| 习题与实践 | 72 |
| 第 3 章 嵌入式 Linux 编程基础 | 74 |
| 3.1 C 语言开发嵌入式系统的优势 | 74 |
| 3.2 GCC 编译器 | 75 |
| 3.2.1 GCC 编译器简介 | 75 |
| 3.2.2 GCC 编译过程 | 76 |
| 3.2.3 GCC 编译器的基本用法 | 79 |
| 3.3 GDB 程序调试 | 82 |
| 3.3.1 GDB 简介 | 82 |

| | | |
|--------|----------------------|-----|
| 3.3.2 | 启动 GDB 的方法 | 82 |
| 3.3.3 | GDB 命令 | 84 |
| 3.4 | makefile 工程管理 | 89 |
| 3.4.1 | makefile 概述 | 89 |
| 3.4.2 | make 程序的命令行选项和参数 | 90 |
| 3.4.3 | makefile 中的依赖关系 | 91 |
| 3.4.4 | makefile 中的规则 | 92 |
| 3.4.5 | makefile 中的宏 | 94 |
| 3.4.6 | makefile 构建多个目标 | 96 |
| 3.4.7 | makefile 隐含规则 | 97 |
| 3.4.8 | makefile 后缀规则 | 97 |
| 3.4.9 | makefile 的模式规则 | 98 |
| 3.4.10 | GNU make 和 GCC 的有关选项 | 100 |
| | 本章小结 | 100 |
| | 习题与实践 | 101 |
| 第 4 章 | 嵌入式 Linux 文件编程 | 103 |
| 4.1 | 文件系统概述 | 103 |
| 4.1.1 | 虚拟文件系统 | 103 |
| 4.1.2 | Linux 文件类型 | 104 |
| 4.1.3 | Linux 文件系统组成 | 106 |
| 4.1.4 | 文件描述符 | 108 |
| 4.2 | Linux 基本文件 I/O 操作 | 108 |
| 4.2.1 | 文件的创建 | 108 |
| 4.2.2 | 文件的打开 | 110 |
| 4.2.3 | 文件的关闭 | 111 |
| 4.2.4 | 读文件 | 112 |
| 4.2.5 | 写文件 | 112 |
| 4.2.6 | 文件定位 | 112 |
| 4.2.7 | 权限判断 | 113 |
| 4.3 | 基于流的标准 I/O 操作 | 115 |
| 4.3.1 | 文件的创建和打开 | 115 |
| 4.3.2 | 读文件 | 116 |
| 4.3.3 | 写文件 | 116 |
| 4.3.4 | 从文件读字符 | 116 |
| 4.3.5 | 向文件写字符 | 117 |
| 4.3.6 | 格式化读 | 117 |
| 4.3.7 | 格式化写入 | 117 |
| 4.3.8 | 文件定位 | 117 |
| 4.3.9 | 获取文件读写位置 | 118 |

| | |
|------------------------------|------------|
| 4.3.10 获取当前路径 | 118 |
| 4.3.11 创建目录 | 118 |
| 4.4 Linux 时间编程 | 120 |
| 4.4.1 时间获取 | 121 |
| 4.4.2 时间转换 | 121 |
| 4.4.3 时间显示 | 122 |
| 4.4.4 取得当前时间 | 123 |
| 4.4.5 延时执行 | 124 |
| 本章小结 | 124 |
| 习题与实践 | 124 |
| 第 5 章 嵌入式 Linux 进程控制 | 126 |
| 5.1 进程控制理论基础 | 126 |
| 5.1.1 进程定义 | 126 |
| 5.1.2 进程特点 | 127 |
| 5.1.3 进程状态 | 127 |
| 5.1.4 进程 ID | 129 |
| 5.1.5 进程互斥 | 129 |
| 5.1.6 临界资源与临界区 | 130 |
| 5.1.7 进程同步 | 130 |
| 5.1.8 进程调度 | 130 |
| 5.1.9 调度算法 | 130 |
| 5.1.10 死锁 | 132 |
| 5.2 进程控制编程 | 132 |
| 5.2.1 获取进程信息 | 132 |
| 5.2.2 进程控制 | 134 |
| 本章小结 | 140 |
| 习题与实践 | 141 |
| 第 6 章 嵌入式 Linux 进程间通信 | 142 |
| 6.1 进程通信概述 | 142 |
| 6.1.1 进程通信目的 | 142 |
| 6.1.2 进程通信发展历程 | 142 |
| 6.1.3 进程通信分类 | 143 |
| 6.2 管道通信 | 143 |
| 6.2.1 管道通信概述 | 143 |
| 6.2.2 无名管道 | 143 |
| 6.2.3 命名管道 | 148 |
| 6.3 信号通信 | 150 |
| 6.3.1 信号概述 | 150 |
| 6.3.2 信号的种类 | 151 |

| | | |
|--------------|------------------------|------------|
| 6.3.3 | 信号的生命周期 | 151 |
| 6.3.4 | 信号相关函数 | 152 |
| 6.4 | 共享内存通信 | 155 |
| 6.4.1 | 共享内存概述 | 155 |
| 6.4.2 | 共享内存操作步骤 | 155 |
| 6.4.3 | 共享内存操作函数 | 156 |
| 6.5 | 消息队列通信 | 160 |
| 6.5.1 | 消息队列概述 | 160 |
| 6.5.2 | 消息队列相关函数 | 160 |
| 6.6 | 信号量通信 | 165 |
| 6.6.1 | 信号量概述 | 165 |
| 6.6.2 | 信号量工作原理 | 165 |
| 6.6.3 | 信号量相关函数 | 166 |
| | 本章小结 | 170 |
| | 习题与实践 | 170 |
| 第 7 章 | 嵌入式 Linux 多线程编程 | 171 |
| 7.1 | 多线程概述 | 171 |
| 7.2 | 多线程程序设计 | 171 |
| 7.3 | 线程属性 | 177 |
| 7.4 | 线程的数据处理 | 185 |
| 7.4.1 | 线程数据 | 185 |
| 7.4.2 | 互斥锁 | 187 |
| 7.4.3 | 条件变量 | 192 |
| 7.4.4 | 信号量 | 195 |
| | 本章小结 | 198 |
| | 习题与实践 | 199 |
| 第 8 章 | 嵌入式 Linux 网络编程 | 201 |
| 8.1 | Linux 系统网络编程概述 | 201 |
| 8.1.1 | Linux 系统网络编程优势 | 201 |
| 8.1.2 | 网络模型 | 201 |
| 8.1.3 | TCP/IP 协议族 | 205 |
| 8.1.4 | TCP/IP 封装 | 206 |
| 8.2 | Linux 网络编程基础 | 211 |
| 8.2.1 | 套接字 | 211 |
| 8.2.2 | 网络地址 | 212 |
| 8.2.3 | 字节序 | 213 |
| 8.2.4 | Socket 编程常用函数 | 214 |
| 8.3 | TCP 网络程序设计 | 217 |
| 8.4 | UDP 网络程序设计 | 221 |

| | |
|--|------------|
| 本章小结 | 225 |
| 习题与实践 | 225 |
| 第 9 章 嵌入式 Linux 系统构建 | 226 |
| 9.1 目标板硬件构建 | 226 |
| 9.1.1 目标板硬件资源 | 226 |
| 9.1.2 目标板外围接口电路原理图 | 229 |
| 9.2 开发环境搭建 | 237 |
| 9.2.1 宿主机开发环境搭建 | 237 |
| 9.2.2 基础软件移植 | 240 |
| 本章小结 | 247 |
| 习题与实践 | 247 |
| 第 10 章 嵌入式 Linux 数据采集系统开发 | 248 |
| 10.1 嵌入式系统开发流程 | 248 |
| 10.2 数据采集系统服务器端软件系统设计 | 249 |
| 10.2.1 服务器端数据采集系统组成 | 249 |
| 10.2.2 Mjpg-streamer 软件移植 | 250 |
| 10.2.3 FFmpeg 移植 | 259 |
| 10.2.4 温度采集程序设计 | 260 |
| 10.2.5 网页交互程序设计 | 263 |
| 10.3 数据采集系统客户端软件设计 | 264 |
| 10.3.1 数据采集系统客户端开发环境的搭建 | 264 |
| 10.3.2 客户端网页设计 | 265 |
| 10.3.3 HTTP 网页请求分析 | 268 |
| 10.3.4 Mjpg-streamer 响应分析 | 270 |
| 10.3.5 基于 Ajax 与 CGI 技术的 HTML 网页设计 | 271 |
| 10.4 B/S 架构嵌入式数据采集系统测试 | 280 |
| 10.4.1 服务器端系统测试 | 280 |
| 10.4.2 客户端系统测试 | 283 |
| 本章小结 | 285 |
| 习题与实践 | 285 |
| 参考文献 | 286 |

第1章 绪 论

嵌入式系统的应用已涉及生产、工作、生活各个方面。从家用电子电器产品中的冰箱、洗衣机、电视、微波炉到 MP3、DVD；从轿车控制到火车、飞机的安全防范；从手机、电话到 PDA；从医院的 B 超、CT 到核磁共振器；从机械加工中心到生产线上的机器人、机械手；从航天飞机、载人飞船到水下核潜艇，到处都有嵌入式系统和嵌入式技术的应用。嵌入式技术无所不在，嵌入式技术和设备的应用在我国国民经济和国防建设的各个方面存在着广泛的应用，有着巨大的市场。可以说它是信息技术的一个新的发展，是信息产业的一个新亮点，也是当前最热门的技术之一。

1.1 嵌入式系统概述

1.1.1 嵌入式系统定义与特点

1. 嵌入式系统的定义

按照历史性、本质性、普遍性要求，嵌入式系统应定义为：嵌入到对象体系中的专用计算机系统。“嵌入性”“专用性”与“计算机系统”是嵌入式系统的三个基本要素。对象系统是指嵌入式系统所嵌入的宿主系统。

根据 IEEE(电气和电子工程师协会)的定义，嵌入式系统是“控制、监视或者辅助装置、机器和设备运行的装置”(devices used to control, monitor, or assist the operation of equipment, machinery or plants)。从中可以看出嵌入式系统是软件和硬件的综合体，还可以涵盖机械等附属装置。

目前国内一个普遍被认同的定义是：以应用为中心，以计算机技术为基础，软硬件可裁剪，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。它一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户的应用程序四部分组成，用于实现对其他设备的控制、监视或管理等功能。

2. 嵌入式系统的特点

从上面的定义可以看出，嵌入式系统具有以下几个重要特征。

(1) 系统内核小。由于嵌入式系统一般应用于小型电子装置，系统资源相对有限，所以其内核比传统的操作系统要小得多。例如，Enea 公司的 OSE 分布式系统，内核只有 5KB，而 Windows 的内核有几吉字节，真是天壤之别。

(2) 专用性强。嵌入式系统的个性化很强，其中的软件系统和硬件的结合非常紧密，一般进行的针对硬件系统的移植，即使在同一系列的产品中也需要根据系统硬件的变化不断进行修改，这种修改和通用软件的升级是完全不同的两个概念。

(3) 系统精简。嵌入式系统一般没有系统软件和应用软件的明显区分，不要求其功能设计

及实现上过于复杂,这样既有利于控制系统成本,也有利于实现系统安全。

(4) 高实时性的系统软件(OS)。嵌入式软件一般具有较高的实时性,并对软件代码的质量和可靠性有较高要求。

(5) 嵌入式软件开发要想走向标准化,就必须使用多任务的操作系统。为了合理地调度多任务,利用系统资源、系统函数以及和专家库函数接口,用户必须自行选配 RTOS(Real Time Operating System),这样才能保证程序执行的实时性、可靠性,并缩短开发时间,保证软件质量。

(6) 嵌入式系统开发需要采用交叉编译的方式。由于宿主机和目标机的体系结构不同,在宿主机(如 X86 平台)上可以运行的程序在目标机(如 ARM 平台)上无法运行,因此嵌入式软件开发采用交叉编译方式在一个平台上生成可以在另一个平台上执行的代码。编译最主要的工作就是将程序转化成运行该程序的 CPU 所能识别的机器代码。进行交叉编译的主机称为宿主机,也就是普通的通用计算机,宿主机系统资源丰富,方便使用各种集成开发环境和调试工具等。程序实际运行的环境称为目标机,也就是嵌入式系统环境。

1.1.2 嵌入式系统应用领域

嵌入式系统无疑是当前最热门、最有发展前途的 IT 应用领域之一。嵌入式系统用在一些特定专用设备上,通常这些设备的硬件资源(如处理器、存储器等)非常有限,并且对成本很敏感,有时对实时响应要求很高。特别是随着消费家电的智能化,嵌入式系统更显重要。像我们平常见到的手机、PDA、电子字典、可视电话、VCD/DVD/MP3 Player、数字相机(DC)、数字摄像机(DV)、机顶盒(Set Top Box)、高清电视(HDTV)、游戏机、智能玩具、交换机、路由器、数控设备或仪表、汽车电子、家电控制系统、医疗仪器、航天航空设备等都是典型的嵌入式系统。

2013~2014 年度的行业调查数据显示,目前嵌入式产品应用最多的三大领域依然是工业控制、消费电子与通信设备,所占比例分别是 25%、17%和 13%,三大领域所占比例之和为 55%,跟 2012 年的情况基本持平。而占据 9%的“其他”一项选择中,参与调查者主要选择的是电力设备、智能电网、物联网、仪器仪表、教育等行业。我们有理由相信,这些都充分表明,未来嵌入式系统将会走进 IT 产业的各个领域,成为推动整个产业发展的核心中坚力量。

当前,人类处在数字信息技术和网络技术高速发展的后 PC 时代,国内外的嵌入式系统已经广泛应用到各个行业、各个学科以及人们的日常生活中。放眼全球,手机、互联网、智能家电、智能交通等嵌入式移动互连已经在世界的各个角落得以应用。在现今的物联网时代,嵌入式系统又面临着新的机遇。

1. 工业控制

嵌入式系统的发展提高了工业控制的自动化程度。随着精密仪器、高精尖技术、生产工艺技术等方面的发展,系统中控制任务可能越来越多且越来越复杂,信息处理往往要经过复杂的算法,因此对嵌入式系统的性能要求越来越高,要求嵌入式系统具有更高的处理能力、更高的可靠性和更强的实时性。

2. 消费电子

嵌入式系统正在越来越广泛地应用于消费电子产品领域,并为之带来了更高的附加值。

如人们日常使用的智能手机、数字电视、便携式多媒体播放器、数码相机、数码相框、可视电话等，嵌入式技术正在人们的生活中占据着越来越重要的地位。在近几年的消费电子产业保持良好的增长势头中，创新成为产业持续高速发展的源动力，是带动和促进消费电子产业升级和产业融合的增长引擎。由3D打印、4G手机、可穿戴设备、曲面超高清电视等更是掀起了消费电子产品创新的热潮。而嵌入式技术成为了创新的主要手段。

3. 网络通信设备

网络通信设备中，嵌入式系统发挥了重要的作用。路由器内部可以划分为控制平面和数据通道，数据通道的主要任务就是数据转发，对于软件转发式路由器来说，系统中的CPU中一定要有一个操作系统来完成一定的工作才能实现软件转发功能。由于路由器的功能相对单一，主要工作就是数据转账，因此嵌入式系统最能适应其工作。在市场份额方面，路由器和交换机占了绝对的比例。2013年，全球电信级路由器和交换机市场创下销售新纪录，总额达到400亿美元。

4. 汽车电子

嵌入式系统在汽车电子中的应用可以分为3个阶段：底层的汽车SCM(Single Chip Microcomputer)系统主要用于任务相对简单、数据处理量小和实时性要求不高的控制场合，如雨刷、车灯系统、仪表盘及电动门窗等；第二代汽车嵌入式系统能够完成简单的实时任务，目前在汽车电控系统中得到了最广泛的应用，如ABS系统、智能安全气囊、主动悬架及发动机管理系统等；第三代汽车SOC系统是嵌入式技术在汽车电子上的高端应用，满足了现代汽车电控系统功能不断扩展、逻辑渐趋复杂、子系统间通信频率不断提高的要求，代表着汽车电子技术的发展趋势，汽车嵌入式SOC系统主要应用在混合动力总成、底盘综合控制、汽车定位导航、车辆状态记录与监控等领域。

5. 军工电子

在20世纪60年代，武器控制中就开始采用嵌入式计算机系统，后来用于军事指挥控制和通信系统，所以军事国防历来就是嵌入式系统的一个重要应用领域。现在各种武器控制(如火炮控制、导弹控制、智能炸弹制导引爆装置)、坦克、舰艇、轰炸机等，陆海空各种军用电子装备、雷达、电子对抗军事通信装备、野战指挥作战用各种专用设备等都可以看到嵌入式系统的影子。应用嵌入式技术的武器在伊拉克战争中就曾经被广泛使用。

6. 信息家电

从技术方式上讲，信息家电是将数字技术和网络技术集成在电冰箱、洗衣机等传统家用电器上。世界各大厂商也正提出许多有关智能家居的解决方案，但业界还没有形成统一的标准，各国正在研究适合本国国情的智能家居系统。而现代嵌入式系统的发展，正好能以低成本、快速的方式满足这些需求。

7. 医疗设备

近年来，越来越多的高科技手段开始运用到医疗仪器的设计中。心电图、脑电图等生理参数检测设备、各种类型的监护仪器、超声波、X射线成影设备、核磁共振仪器以及各式各样的物理治疗仪都开始在各地医院广泛使用。从庞大的要占用一整间房的核磁共振成像扫描仪到便携式和手持仪器，再到如心脏起搏器等植入式设备都采用嵌入式系统。

另外，医疗仪器作为一个特殊的行业，又要求设备能够达到更高级别的环保要求。如何进一步智能化、专业化、小型化，同时做到低功耗、零污染，将会是一个无止境追求过程，这为嵌入式系统在医疗仪器中的应用提供了更广阔的天地和更高的要求。

8. 航空航海

在航空航海领域，嵌入式系统可以作为火箭发射的主控系统，可以作为卫星信号测控系统，可以是飞机上的飞控系统，也可以做成瞄准系统，还可以作为水下航行器的控制系统。例如，小型水下航行器嵌入式控制系统可用于探测海底地貌和资源，采集信息和数据处理等，而且能对航行器的姿态、航速、航深及航行路线进行自动控制。

1.1.3 嵌入式系统发展历程与发展趋势

嵌入式系统的出现最初是基于单片机的。从 20 世纪 70 年代单片机出现到今天各式各样的嵌入式微控制器、微处理器的大规模应用，嵌入式系统少说也有近 40 年的历史。纵观嵌入式系统的发展历程，大致经历了以下四个阶段。

1. 无操作系统阶段

嵌入式系统最初的应用是基于单片机的，大多以可编程控制器的形式出现，具有监测、伺服、设备指示等功能，通常应用于各类工业控制、家电、汽车电子和武器装备中，一般没有操作系统的支持，只能通过汇编语言对系统进行直接控制。这些装置已经初步具备了嵌入式的应用特点，但是这时的应用只是使用 8 位的芯片，执行一些单线程的程序，还谈不上系统的概念。

这个阶段嵌入式系统的主要特点是：系统结构和功能相对单一，处理效率较低，存储容量较小，几乎没有用户接口。由于这种嵌入式系统功能简单、价格低廉，因而在工业控制领域得到了十分广泛的应用。

2. 简单操作系统阶段

20 世纪 80 年代，随着微电子工艺水平的不断提高，集成电路制造商开始把嵌入式应用中所需要的微处理器、I/O 接口、串行接口以及 RAM、ROM 等部件全部集成到一片超大规模集成电路中，制造出面向 I/O 设计的微控制器，并一举成为嵌入式系统领域中异军突起的新秀。与此同时，嵌入式系统的程序员开始用商业级的操作系统编写嵌入式应用软件，这使得可以获取更短的开发周期、更低的开发资金和更高的开发效率，出现了真正意义上的嵌入式系统。确切地说，这个时候的操作系统是一个实时核，这个实时核包含了许多传统操作系统的特征，包括任务管理、任务间通信、同步与互斥、中断支持、内存管理等功能。其中有代表性的有 Ready System 公司的 VRTX、ISI(Integrated System Incorporation)的 PSOS、IMG 的 VxWorks、QNX 公司的 QNX 等。这些嵌入式操作系统都具有嵌入式的典型特点：它们均采用占先式调度，响应时间很短，任务执行时间可以确定，实时性较强；系统内核很小，可裁剪、扩充和移植，并且能够移植到各种不同体系架构的微处理器上；可靠性高，适合于嵌入式应用。这些嵌入式实时多任务操作系统的出现，扩大了应用开发人员的开发范围，同时使嵌入式系统有了更为广阔的应用空间。

这个阶段嵌入式系统的主要特点是：出现了大量可靠性高、功耗低的嵌入式 CPU(如 PowerPC 等)，各种简单的嵌入式操作系统开始出现并得到迅速发展。相应的嵌入式操作系统

虽然比较简单,但已经初步具备了一定的兼容性和扩展性,内核精巧且效率高,主要用来控制系统负载和监控应用程序的运行。

3. 实时操作系统阶段

20世纪90年代,在分布式控制、柔性制造、数字化通信和信息家电等巨大需求的引领下,嵌入式系统进一步飞速发展,面向实时信号处理算法的DSP产品向着高速、高精度、低功耗的方向发展。随着硬件实时性要求的提高,嵌入式系统的软件规模也不断扩大,逐渐形成了实时多任务操作系统,并开始成为嵌入式系统的主流。

这时候更多的公司看到了嵌入式系统的广阔发展前景,开始大力发展自己的嵌入式操作系统,出现了Palm OS、WinCE、嵌入式Linux、Lynx、Nucleux,以及国内的Hopen、DeltaOs等嵌入式操作系统。

这个阶段嵌入式系统的主要特点是:操作系统的实时性得到了很大改善,已经能够运行在各种不同类型的微处理器上,具有高度的模块化和扩展性。此时的嵌入式操作系统已经具备了文件和目录管理、设备管理、多任务、网络、图形用户界面等功能,并提供了大量的应用程序接口,从而使得应用软件的开发变得更加简单。

4. 面向Internet阶段

近年来,随着通信技术、网络技术和半导体技术的飞速发展,Internet技术正在逐渐向工业控制和嵌入式系统设计领域渗透,嵌入式系统的设计步入了崭新的时代。实现Internet互连是当前嵌入式系统发展的重要方向,通过为现有嵌入式系统增加Internet接入能力来扩展其功能,使得家用电器、工业控制装置或仪器、安全监控系统、汽车电子等各种智能设备的Internet互连成为可能。

嵌入式技术与Internet技术的结合正推动着嵌入式技术的飞速发展,嵌入式系统的研究和应用产生了显著变化,这个阶段嵌入式系统的主要特点如下。

(1)新的微处理器层出不穷,嵌入式操作系统自身结构的设计更加便于移植,能够在短时间内支持更多的微处理器。

(2)嵌入式系统的开发成了一项系统工程,开发厂商不仅要提供嵌入式软硬件系统本身,还要提供强大的硬件开发工具和软件支持包。

(3)通用计算机上使用的新技术、新观念开始逐步移植到嵌入式系统中,如嵌入式数据库、移动代理、实时CORBA等,嵌入式软件平台得到进一步完善。

(4)各类嵌入式Linux操作系统迅速发展,由于具有源代码开放、系统内核小、执行效率高、网络结构完整等特点,很适合信息家电等嵌入式系统的需要,目前已经形成了能与Windows CE、Palm OS等嵌入式操作系统进行有力竞争的局面。

(5)网络化、信息化的要求随着Internet技术的成熟和带宽的提高而日益突出,以往功能单一的设备(如电话、手机、冰箱、微波炉等)功能不再单一,结构变得更加复杂,网络互连成为必然趋势。

未来嵌入式系统的主要发展趋势有以下几方面。

(1)小型化、智能化、网络化、可视化。随着电子技术、网络技术和人们生活需求的不断提高,嵌入式设备(尤其是消费类产品)正朝着小型化、便携式和智能化的方向发展,如目前的上网本、MID(移动互联网设备)、便携投影仪等。对嵌入式而言,可以说已经进入

了嵌入式 Internet 时代(有线网、无线网、广域网、局域网的组合), 嵌入式设备和 Internet 的紧密结合更为人们的日常生活带来了极大的方便。嵌入式设备功能越来越强大, 未来冰箱、洗衣机等家用电器都将实现网上控制; 异地通信、协同工作、无人操控场所、安全监控场所等的可视化也已经成为现实, 而且随着网络运载能力的提升, 可视化将得到进一步完善。人工智能、模式识别技术也将在嵌入式系统中得到应用, 使得嵌入式系统更加人性化、智能化。

(2) 多核技术的应用。人们需要处理的信息越来越多, 这就要求嵌入式设备运算能力更强, 因此需要设计出更强大的嵌入式处理器, 多核技术处理器在嵌入式中的应用将更为普遍。

(3) 低功耗、绿色环保。在嵌入式系统的硬件和软件设计中都在追求更低的功耗, 以求嵌入式系统能获得更长的可靠工作时间, 如手机的通话和待机时间、MP3 听音乐的时间等。同时, 绿色环保型嵌入式产品将更受人们的青睐, 在嵌入式系统设计中也会更多地考虑辐射和静电等问题。

(4) 云计算、可重构、虚拟化等技术被进一步应用到嵌入式系统中。简单地讲, 云计算是将计算分布在大量的分布式计算机上, 这样只需要一个终端, 就可以通过网络服务来实现需要的计算任务, 甚至是超级计算任务。云计算(Cloud Computing)是分布式处理(Distributed Computing)、并行处理(Parallel Computing)和网格计算(Grid Computing)的发展, 在未来几年里, 云计算将得到进一步的发展与应用。

可重构性是指在一个系统中, 其硬件模块或(和)软件模块均能根据变化的数据流或控制流对系统结构和算法进行重新配置(或重新设置)。可重构系统最突出的优点就是能够根据不同的应用需求改变自身的体系结构, 以便与具体的应用需求相匹配。

虚拟化是指计算机软件在一个虚拟的平台上而不是在真实的硬件上运行。虚拟化技术可以简化软件的重新配置过程, 易于实现软件的标准化。其中 CPU 的虚拟化可以实现单 CPU 模拟多 CPU 并行运行, 允许一个平台同时运行多个操作系统, 并且都可以在相互独立的空间内运行而互不影响, 从而提高工作效率和安全性。虚拟化技术是降低多内核处理器系统开发成本的关键。虚拟化技术是未来几年最值得期待和关注的关键技术之一。

(5) 嵌入式软件开发平台化、标准化、系统可升级、代码可复用将更受重视。嵌入式操作系统将进一步开放、开源、标准化、组件化。嵌入式软件开发平台化也将是今后的一个趋势, 越来越多的嵌入式软硬件行业标准将出现, 最终的目标是使嵌入式软件开发简单化, 这也是一个必然规律。同时随着系统复杂度的提高, 系统可升级和代码复用技术在嵌入式系统中得到更多应用。另外, 因为嵌入式系统采用的微处理器种类多, 且不够标准, 所以在嵌入式软件开发中将更多地使用跨平台的软件开发语言与工具。目前, Java 语言正在被越来越多地应用到嵌入式软件开发中。

(6) 嵌入式系统软件将逐渐 PC 化。移动互联网的发展, 将进一步促进嵌入式系统软件 PC 化。如前所述, 结合跨平台开发语言的广泛应用, 未来嵌入式软件开发的观念将被逐渐淡化, 也就是说, 嵌入式软件开发和非嵌入式软件开发的区别将逐渐减小。

(7) 融合趋势。嵌入式系统软硬件融合、产品功能融合、嵌入式设备和互联网的融合趋势加剧。嵌入式系统设计中软硬件结合将更加紧密, 软件将是其核心。消费类产品将在运算能力和便携方面进一步融合。传感器网络的迅速发展, 将极大地促进嵌入式技术和互联网技术的融合。

(8) 安全性。随着嵌入式技术和互联网技术的结合发展, 嵌入式系统的信息安全问题日益凸显, 保证信息安全也成为嵌入式系统开发的重点和难点。

1.1.4 嵌入式系统组成

嵌入式系统结构分为五层，即硬件层、驱动层、操作系统层、中间件层和应用软件层，如图 1-1 所示。

1. 硬件层

硬件层是嵌入式系统的基础。嵌入式系统硬件层一般包括嵌入式处理器、存储器(SDRAM、ROM、Flash 等)、I/O 接口(A/D、D/A、串口、并口、USB 口等)、输入/输出设备(键盘、LCD 等)和系统总线。

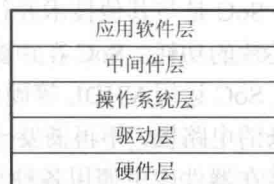


图 1-1 嵌入式系统结构

1) 嵌入式处理器

嵌入式处理器是嵌入式系统硬件层的核心部件，分为嵌入式微控制器(Embedded Microcontroller Unit, EMCU)、嵌入式 DSP 处理器(Embedded Digital Signal Processor, EDSP)、嵌入式微处理器(Embedded Micro Processor Unit, EMPU)和嵌入式片上系统(Embedded System on Chip, ESoC)等几类。

(1) 嵌入式微控制器。嵌入式微控制器是将整个计算机系统集成到一个芯片中，芯片内部集成 ROM/E²PROM、RAM、总线、总线逻辑、定时器/计数器、看门狗、I/O、串行口、脉宽调制输出、A/D、D/A、Flash RAM 等各种必要的功能和外设，由于其片上外设资源比较丰富，适合于控制，故称微控制器，其典型代表是单片机。

嵌入式微控制器的最大特点是单片化、体积小、功耗和成本低、可靠性高，比较有代表性的包括 51 系列、AVR 系列、PIC16 系列等。EMCU 目前仍然被广泛应用，并占嵌入式系统约 70% 的市场份额。

(2) 嵌入式 DSP 处理器。嵌入式 DSP 处理器是用于信号处理方面的专用处理器，由于其在系统结构和指令算法方面进行了特殊设计，因而具有很高的编译效率和指令执行速度，并在数字滤波、FFT、谱分析等各种仪器上得到广泛应用。

最为广泛应用的是 TI 的 TMS320C2000/C5000/C6000 系列，另外如 Intel 的 MCS-296 和 Siemens 的 TriCore 也有各自的应用范围。EDSP 的应用趋势是和微处理器的融合，如 TI 的 OMAP3630(用于 MOTO ME525+手机)，就是把一个 ARM 的 Cortex A8 内核和一个 DSP 内核 C64X 集成在一个芯片上，同时意味着控制和数字信号的结合。

(3) 嵌入式微处理器。嵌入式微处理器是由通用计算机中的 CPU 演变而来的。在应用设计中，将嵌入式微处理器装配在专门设计的电路板上，只保留和嵌入式应用有关的母版功能，这样可以大幅度减小系统体积和功耗。为了满足嵌入式应用的特殊要求，嵌入式微处理器虽然在功能上和标准微处理器基本是一样的，但在工作温度、抗电磁干扰、可靠性等方面一般都作了各种增强设计。

嵌入式微处理器的体系主要取决于它所采用的存储结构和指令系统。有很多种不同的嵌入式微处理器体系，即使是在同一体系中也具有不同的时钟速度和总线数据宽度，集成不同的外部接口和设备。据不完全统计，目前全世界嵌入式微处理器的品种数量已经超过千种，有几十种嵌入式微处理器体系，主流的体系包括 ARM、MIPS、PowerPC、X86 等。

(4) 嵌入式片上系统。嵌入式片上系统指的是在单个芯片上集成一个完整的系统，对所有或部分必要的电子电路进行包分组的技术。SoC 追求系统最大限度的集成，其最大特点是在