高等院校计算机专业系列规划教材

GAODENG YUANXIAO JISUANJI ZHUANYE XILIE GUIHUA JIAOCAI

嵌入式系统软件设计

谢荣生 主 编

丰 海 李远敏 卢俊文 副主编

QIANRUSHI XITONG RUANJIAN SHEJI

北京邮电大学出版社
www.buptpress.com

高等院校计算机专业系列规划教材

嵌入式系统软件设计

谢荣生　主　编
丰　海　李远敏　卢俊文　副主编

北京邮电大学出版社
·北京·

内 容 简 介

本书基于 ARM 处理器和嵌入式 Linux 开发环境，全面讲述了嵌入式系统软件开发流程及主要内容。本书分为五大部分，共 10 章。第一部分包括第 1、2 章，讲述嵌入式 Linux 开发环境的搭建和嵌入式 Linux 软件开发基础；第二部分包括第 3、4、5 章，分别讲述嵌入式 BootLoader、内核和根文件系统的基础理论及其移植；第三部分为第 6 章，讲述嵌入式系统驱动程序设计；第四部分为嵌入式应用程序设计，包括第 7、8、9 章，主要讲述基于 Qt 的嵌入式 GUI 设计和嵌入式数据库程序设计；第五部分为第 10 章，通过两个实际的嵌入式软件开发项目，加深对前述内容的理解，提高综合应用能力。

本书可作为高等院校的计算机、电子类相关专业嵌入式系统相关课程的教科书，也可作为基于 ARM 核嵌入式系统软件开发的工程技术人员的参考资料。

图书在版编目(CIP)数据

嵌入式系统软件设计/谢荣生主编. --北京：北京邮电大学出版社，2011.1

ISBN 978-7-5635-2496-9

Ⅰ. ①嵌… Ⅱ. ①谢… Ⅲ. ①微处理器—系统设计 Ⅳ. ①TP332

中国版本图书馆 CIP 数据核字(2010)第 264135 号

书　　名：嵌入式系统软件设计

主　　编：谢荣生

责任编辑：张珊珊

出版发行：北京邮电大学出版社

社　　址：北京市海淀区西土城路 10 号(邮编：100876)

发 行 部：电话：010-62282185　传真：010-62283578

E-mail：publish@bupt.edu.cn

经　　销：各地新华书店

印　　刷：北京忠信诚胶印厂

开　　本：787 mm×1 092 mm　1/16

印　　张：19.25

字　　数：504 千字

印　　数：1—3 000 册

版　　次：2011 年 1 月第 1 版　2011 年 1 月第 1 次印刷

ISBN 978-7-5635-2496-9　　定　价：35.00 元

前　言

目前我国处于工业化中期，面临信息化的重大发展机遇。党的十七大报告明确指出：要加快推进工业化和信息化的融合。只有以信息化带动工业化，以工业化促进信息化，推进工业化和信息化融合，才能走出一条“科技含量高、经济效益好、资源消耗低、环境污染少、人力资源优势得到充分发挥”的新型工业化道路。

在我国，近年来以电子与计算机技术为主导的信息化技术已得到了长足的发展，信息产业已成为国民经济的重要支柱产业，移动电话、程控交换机、计算机等产品的产量位居世界前列。但同时也存在着信息产业大而不强，自主创新能力较弱，核心技术和关键装备受制于人的问题。而作为全球制造业中心，我国制造业面临同样困难与挑战。要推动信息产业结构优化升级，加快推进信息化与工业化的融合，推进新型工业化进程，打造先进制造业基地。

在实现信息化与工业化融合的具体途径方面，以芯片和软件为核心的嵌入式系统技术将发挥巨大作用，将起到连接信息化与工业化的“桥梁”作用。嵌入式系统“嵌入”到工业产品与现代制造装备产品中，将有效提升产品的性价比与市场竞争力，典型的应用包括汽车电子、数控加工装备、智能仪器仪表、自动化控制系统等。由于不掌握嵌入式系统技术，很多制造厂商利润低下，濒临淘汰的境地。

因此，发展嵌入式技术及其在先进制造领域中的应用，设计面向领域的嵌入式产品是促进信息化与工业化融合的有效手段，也是实现我国信息产业与制造业产业调整与产品升级换代，提升我国信息产业的国际竞争力，完成我国从“制造大国”向“制造强国”转变的必由之路。

从开发的角度，嵌入式系统开发包括软件开发和硬件开发两部分。目前，从应用的角度，全面而又系统地讲述嵌入式系统软件设计、开发的教材并不多，这促使我们编写了该教材。

《嵌入式系统软件设计》是一门具有较高综合性的理论和实践相结合的课程。从涉及的内容上看，主要包括：嵌入式软件开发环境；嵌入式 GUI 设计；嵌入式 Bootloader 设计；嵌入式内核设计；嵌入式根文件系统设计；嵌入式应用程序设计；嵌入式软件工程等。全书共 10 章，分为五部分：嵌入式 Linux 开发环境和开发基础、嵌入式软件平台、嵌入式驱动程序设计、嵌入式应用程序设计、嵌入式系统软件开发应用实例，如图 P-1 所示。

第 1 章介绍嵌入式交叉开发环境、嵌入式软件开发过程和主要开发内容，重点讲解了嵌入式 Linux 开发换将构建的方法和步骤。本章的学习为后续章节知识的学习和应用打下必不可少的基础。

第 2 章介绍嵌入式 Linux 开发基础，主要包括交叉编译器和交叉编译、动态库和静态库的制作和使用、嵌入式交叉调试、汇编和 C 的混合编程、嵌入式 Linux 多线程程序设计、嵌入式系统网络程序设计等。

第 3 章介绍嵌入式 BootLoader 基本原理，并以 BLOB 为例介绍了 BLOB 的移植、开发。

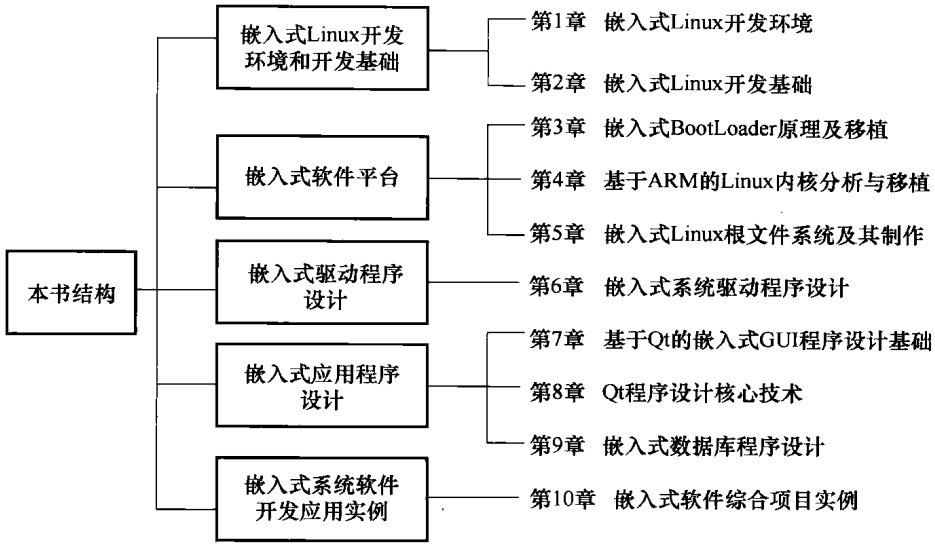

图 P-1 本书纲要

第 4 章在分析嵌入式 Linux 内核的基础上，着重介绍了嵌入式 Linux 内核的裁剪和移植方法、步骤。

第 5 章介绍了嵌入式 Linux 根文件系统结构和制作嵌入式根文件系统的工具 BusyBox。进而，重点讲述了使用 BusyBox 生成、移植的方法和步骤。

ARM Linux 设备驱动程序是嵌入式系统软件开发中非常重要的组成部分。Linux 设备驱动程序通常分为 3 种类型：字符设备、块设备和网络设备。第 6 章重点讲述这 3 类设备驱动程序设计的开发。

第 7 章分析了嵌入式 GUI 的特点，介绍了常用的几种嵌入式 GUI。重点介绍了 Qt 及其主要开发工具，进而讲述了 Qt 开发环境的搭建方法、步骤及 Qt/Embedded 应用程序的编译和运行。

第 8 章讲述 Qt 程序设计涉及的核心技术，主要包括 Qt 对象树、信号和槽机制、Qt 对象模型、基于 Qt 设计器的程序设计、Qt 布局管理、Qt 国际化、Qt 标准对话框和消息框等。

第 9 章介绍了嵌入式数据库的内涵、特征和几种典型的嵌入式数据库。重点讲述 SQLite 数据库及其移植和应用。

第 10 章介绍两个嵌入式软件系统设计的实际项目，通过这两个项目的介绍加深对嵌入式软件开发所涉及的主要内容、主要流程的理解。

本书的作者均是教学和科研一线的骨干教师，具有多年的嵌入式系统教学和开发经验。本书的部分内容取材于作者的嵌入式系统开发科研项目。本书内容系统全面而又重点突出，阐述循序渐进、由浅入深。各章均安排了丰富的习题，便于学生自学和自测。

本书由谢荣生策划和统稿。李远敏撰写了第 1、2 章；卢俊文撰写了 3、4 章；丰海撰写了 5、6 章；谢荣生撰写了 7、8、9、10 章。在本书撰写过程中，引用了参考文献所列论著的有关部分，在此向论著作者一并表示衷心的感谢。

本书为厦门理工学院教材建设基金资助项目。

由于作者水平有限，书中不妥之处在所难免，恳请各位专家和读者批评指正。

编 者

目　录

第1章　嵌入式Linux开发环境

构建嵌入式软件开发环境是从事嵌入式软件开发的前提和必要条件，本身也是嵌入式软件开发的首要内容。

嵌入式Linux系统由于其支持多种硬件平台、开源、成熟稳定、功能强大、资源丰富、可移植性好等诸多优点而成为嵌入式产品开发首选的操作系统。

本章介绍嵌入式交叉开发环境、嵌入式软件开发过程和主要开发内容，重点讲解了嵌入式Linux开发环境构建的方法和步骤。本章的学习为后续章节知识的学习和应用打下必不可少的基础。

1.1　嵌入式交叉开发环境

嵌入式系统通常是一个资源受限的专用计算机系统，直接在嵌入式系统的硬件平台上编写、编译软件比较困难，有时甚至是不可能的，这就引出了交叉编译的概念。所谓交叉编译，简单地说，就是在一个平台上编译、生成另一个平台上的可执行代码。交叉开发环境是指用于嵌入式软件开发的所有工具软件的集合，一般包括文本编辑器、交叉编译器、交叉调试器、仿真器、下载器等。

交叉开发环境由宿主机和目标机组成，如图1.1所示。

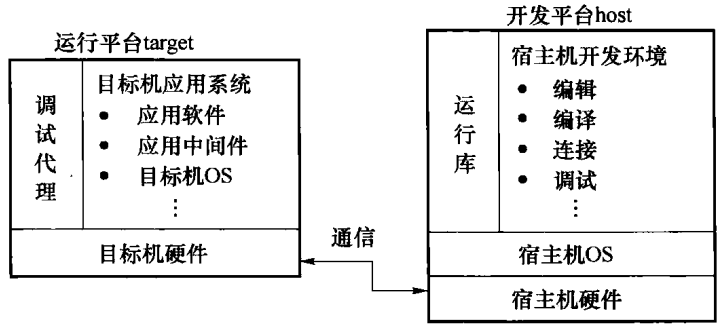

图1.1　嵌入式交叉开发环境

(1) 宿主机(host)：编辑和编译程序的平台，一般是基于X86的PC，通常也被称为主机。宿主机具备丰富的软硬件资源，为嵌入式软件的开发提供全过程支持。

(2) 目标机(target)：所开发的嵌入式系统是嵌入式软件的运行环境，其软硬件是为特定应用定制的。host编译得到的可执行代码在target上运行。

在开发过程中，目标机端需接收和执行宿主机发出的各种命令如设置断点、读内存、写内存等，将结果返回给宿主机，配合宿主机各方面的工作。因而宿主机与目标机之间需要进行有

效的通信。宿主机与目标机之间的通信连接分为物理连接和逻辑连接两种。

(1) 物理连接是指宿主机与目标机通过物理线路连接在一起，是逻辑连接的基础。连接方式主要有三种：

- 串口；
- 以太口；
- OCD(On Chip Debug)方式，如 JTAG。

这三种连接方式都非常重要，一般嵌入式系统的开发都会用到。后面嵌入式开发环境的构建中将详细介绍这些连接的应用。

(2) 逻辑连接指宿主机与目标机间按某种通信协议建立起来的通信连接，目前逐步形成了一些通信协议的标准。

1.2 嵌入式软件开发的过程

嵌入式软件开发的过程主要包括三个步骤：生成、调试和固化运行。

1.2.1 嵌入式软件的生成

嵌入式软件的生成主要是在宿主机上进行，利用各种工具完成对应用程序的编辑、交叉编译和链接工作，生成可供调试或固化的目标程序。主要包括三个过程(如图 1.2 所示)：

- 源代码程序的编写；
- 编译成各个目标模块；
- 链接成可供下载调试或固化的目标程序。

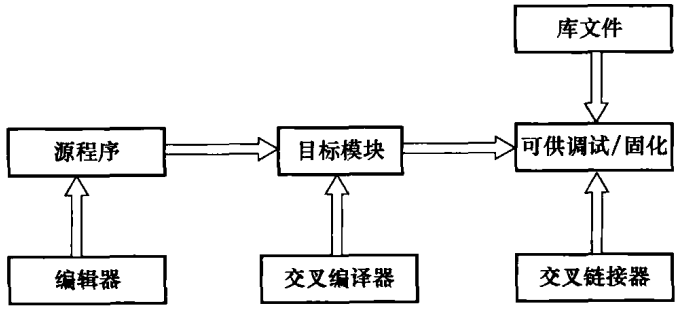

图 1.2 嵌入式软件的生成

1.2.2 嵌入式软件的调试

嵌入式软件的调试是通过交叉调试器来完成的，调试完成后还需进行必要的测试工作。交叉调试器是指调试程序和被调试程序运行在不同机器上的调试器，调试器通过某种方式能控制目标机上被调试程序的运行方式，通过调试器能查看和修改目标机上的内存、寄存器以及被调试程序中的变量等。主要的交叉调试方式有以下几种。

1. Crash and Burn

Crash and Burn 是最早的嵌入式应用软件调试方法，也是最简单的。这种方式主要靠程序员的主观判断。图 1.3 是这种方式的示意图。

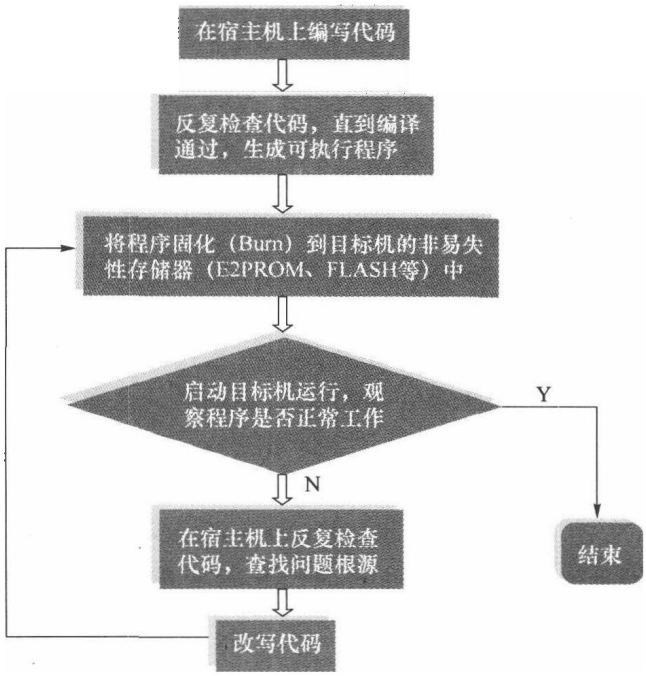

图 1.3　Crash and Burn 调试

2. ROM Monitor

ROM Monitor 是被固化且运行在目标机上的一段程序，负责监控目标机上被调试程序的运行，与宿主机端的调试器一起完成对应用程序的调试。调试器与 ROM Monitor 之间的通信遵循远程调试协议。这种调试方式如图 1.4 所示。

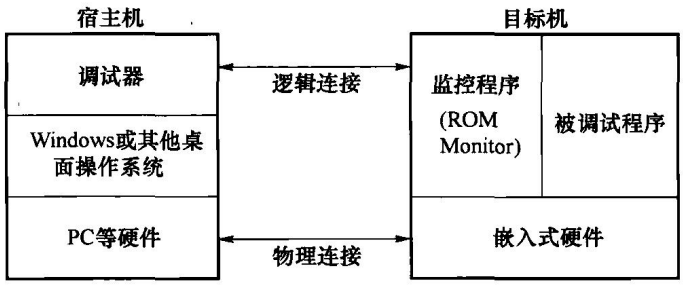

图 1.4　ROM Monitor 调试

这种方式的主要优点是：

- 提高调试程序的效率，缩短开发周期，降低成本；
- 简单、方便；
- 可扩展性强，可支持许多高级调试功能；
- 成本低廉，不需专门的调试硬件支持；
- 几乎所有的交叉调试器都支持这种方式。

缺点是：

- Debug Monitor 需要用 Crash and Burn 方法开发；
- 当 ROM Monitor 占用 CPU 时，应用程序不响应外部的中断，因此不便调试有时间特性的程序；

- ROM Monitor 要占用目标机一定数量的资源，如 CPU、RAM、ROM 和通信设备等资源；
- 调试环境不同于实际目标环境。

3. ROM Emulator

ROM Emulator 是一种用于替代目标机上的 ROM 芯片的设备，即 ROM 仿真器。利用这种设备，目标机可以没有 ROM 芯片，但目标机的 CPU 可以读取 ROM Emulator 设备上 ROM 芯片的内容：ROM Emulator 设备上的 ROM 芯片的地址可以实时地映射到目标机的 ROM 地址空间，从而仿真(Emulation)目标机的 ROM。

ROM Emulator 的调试方式是一种不完全的调试方式：ROM Emulator 设备只是为目标机提供 ROM 芯片及在 Target 和 Host 间建立一条高速的通信通道，因此它经常和前面两种调试方式结合起来形成一种完备的调试方式。ROM Emulator 的典型应用就是和 ROM Monitor 的调试方式相结合。

这种调试方式的优点是：保证调试版本与最终发布版本一致。缺点是：目标机必须能支持外部 ROM 存储空间，而且由于其通常要和 ROM Monitor 配合使用，因此它拥有 ROM Monitor 的缺点。

4. ICE

ICE(In-Circuit Emulator)是一种用于替代目标机上 CPU 的设备，即在线仿真器。它比一般的 CPU 有更多的引出线，能够将内部的信号输出到被控制的目标机。ICE 上的 Memory 也可以被映射到用户的程序空间，这样即使在目标机不存在的情形下也可以进行代码的调试。

连接 ICE 和目标机时，一般是将目标机的 CPU 取下，而将 ICE 的 CPU 引出线接到目标机的 CPU 插槽。用 ICE 进行调试时，在 Host 端运行的调试器通过 ICE 来控制目标机上运行的程序。

这种方式能够同时支持软件断点和硬件断点的设置，设置各种复杂的断点和触发器，实时跟踪目标程序的运行，并可实现选择性的跟踪。因而，特别适用于调试设备驱动程序、实时的应用系统，适合对硬件进行功能和性能的测试。主要缺点是价格太昂贵，不利于团队开发，所仿 CPU 有限。

5. OCD

OCD(On Chip Debugging)是 CPU 芯片提供的一种调试功能(片上调试)，可以认为是一种廉价的 ICE 功能：OCD 的价格只有 ICE 的 20%，但提供了 ICE 80%的功能。OCD 调试结构如图 1.5 所示。

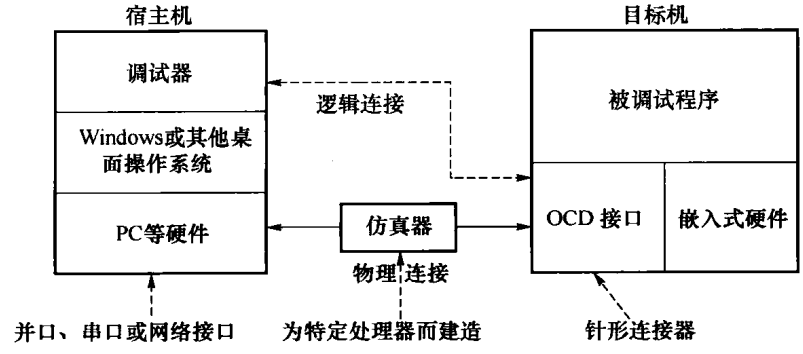

图 1.5　OCD 调试结构

这种调试方式的优点是：

- 不占用目标机的资源；
- 调试环境和最终的程序运行环境基本一致；
- 支持软硬断点、Trace 功能；
- 精确计量程序的执行时间；
- 提供时序分析功能。

缺点是：

- 调试的实时性不如 ICE；
- 不支持非干扰调试查询；
- CPU 必需具有 OCD 功能。

OCD 的实现方式主要有 BDM(Background Debugging Mode)、JTAG(Joint Test Access Group)、OnCE(On Chip Emulation)等几种。其中，JTAG 是主流方式。

1.2.3 嵌入式软件的固化

固化运行是先用一定的工具将程序固化到目标机上，然后启动目标机，在没有任何工具干预的情况下程序能自动地启动运行。根据目标程序的不同，固化的方式可能不一样。比如，Bootloader 的固化一般采用 JTAG 接口，而内核、根文件系统和应用程序适用于速度更快的网口。

1.3 嵌入式 Linux 软件开发的主要内容

嵌入式 Linux 可以运行的硬件平台十分广泛，从 X86、MIPS、POWERPC 到 ARM，以及其他许多硬件体系结构。目前在世界范围，ARM 体系结构的 SOC 逐渐占领 32 位嵌入式微处理器市场，并且在国内市场上很容易购买到 ARM 核的嵌入式处理器。所以，本书所涉及的嵌入式软件开发针对的是嵌入式 Linux 软件平台和 ARM 硬件平台。所采用的 Linux 版本是 2.6.9；书中的实验采用的硬件是 Up-techpxa270-s 实验平台，当然，所介绍的嵌入式软件开发方法同样适用于其他基于嵌入式 Linux 系统的硬件环境。

基于 Linux 环境的嵌入式软件的开发，从整体上看包括以下几个方面的内容，这些开发内容其实也是承前启后的。

1. 建立开发环境

主要包括以下几个方面的内容。

(1) 在宿主机上安装 Linux 操作系统，现在一般安装 REDHAT 9。

(2) 通过网络下载相应的 gcc 交叉编译器进行安装(比如 arm-linux-gcc、arm-uclibc-gcc)，或者安装产品厂家提供的交叉编译器。

(3) 配置开发主机，主要包括：①配置 Minicom，或者在 Windows 下配置超级终端，Minicom 软件的作用是作为调试嵌入式开发板的信息输出的监视器和键盘输入的工具；②配置网络，包括配置 NFS 网络文件系统、tftp 服务等。

2. 开发引导装载程序 BOOTLOADER

独立开发 BOOTLOADER 是一件非常艰巨的工作，就像独立开发内核一样。因此，一般的做法是从网络上下载一些公开源代码的 BOOTLOADER，如 U-BOOT、BLOB、VIVI、LILO、ARM-BOOT、RED-BOOT 等，根据自己的具体芯片进行移植修改。

3. 移植嵌入式 Linux 操作系统

获取 Linux 内核源码，针对系统的硬件平台进行移植。如果有专门针对用户所使用的 CPU 移植好的 Linux 操作系统那是再好不过。移植过程中要添加自己的特定硬件的驱动程序，进行调试修改，对于带 MMU 的 CPU 可以使用模块方式调试驱动，对于 uCLinux 这样的系统只能编译进内核进行调试。

4. 建立根文件系统

从 www. busybox. net 下载实用 BUSYBOX 软件进行功能裁剪，产生一个最基本的根文件系统，再根据自己的应用需要添加其他的程序。默认的启动脚本一般都不会符合应用的需要，所以就要修改根文件系统中的启动脚本，它的存放位置位于/etc 目录下，包括/etc/init. d/rc. S、/etc/profile、/etc/. profile 等，以及自动挂装文件系统的配置文件/etc/fstab，具体情况会随系统不同而不同。根文件系统在嵌入式系统中一般设为只读，需要使用 mkcramfs、genromfs 等工具产生烧写映像文件。

5. 建立应用程序的 FLASH 磁盘分区

一般使用 JFFS2 或 YAFFS 文件系统，这需要在内核中提供这些文件系统的驱动，有的系统使用一个线性 FLASH(NOR 型，512 KB～32 MB)，有的系统使用非线性 FLASH(NAND 型，8 MB～512 MB)，有的两个同时使用，需要根据应用规划 FLASH 的分区方案。

6. 开发应用程序

应用程序可以放入根文件系统中，也可以放入 YAFFS、JFFS2 文件系统中，有的应用不使用根文件系统，直接将应用程序和内核设计在一起，这有点类似于 UCOS - Ⅱ的方式。

7. 系统固化

固化内核、根文件系统、应用程序。

1.4 构建嵌入式 Linux 开发环境

本书将对嵌入式 Linux 软件开发所涉及的主要内容进行全面介绍，而搭建嵌入式交叉开发环境是嵌入式开发的第一步，也是必备一步。接下来介绍嵌入式 Linux 开发环境的构建，这也是本章的重点。

1.4.1 开发平台 Linux 操作系统的安装

目前，嵌入式 Linux 开发环境有几个方案：

(1) 基于 PC 的 Windows 操作系统下的 CYGWIN；

(2) 直接安装 Linux 操作系统；

(3) 在 Windows 下安装虚拟机后，再在虚拟机中安装 Linux 操作系统。

由于开发人员对 Windows 系统比较熟悉，同时为了充分利用 Windows 平台下资源，一般把 Linux 操作系统安装在 Windows 下的虚拟机上，用得比较多的虚拟机是 Vmware。

关于虚拟机 Vmware 和 Linux 系统的安装，本书不作讲述，读者可参考其他参考书。

1.4.2 嵌入式交叉编译环境的搭建

搭建交叉编译环境的方法很多，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，而且，有些交叉编译器经常会有部分的 BUG，这都会导致最后的代码无法正常地运行。因此，选择合适的交叉编译器对于嵌入式开发是非常重要的。