

## <<嵌入式系统原理及开发>>

### 图书基本信息

书名：<<嵌入式系统原理及开发>>

13位ISBN编号：9787121139246

10位ISBN编号：7121139243

出版时间：2011-7

出版时间：电子工业出版社

作者：应忍冬

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<嵌入式系统原理及开发>>

### 内容概要

这本《嵌入式系统原理及开发——基于ATOM处理器及Linux实现》由应忍冬、蒋乐天、徐国治编著，主要介绍嵌入式系统软硬件架构技术及嵌入式系统优化技术。

内容涵盖了嵌入式系统从底层到上层的软硬件设计技术，在硬件方面分别介绍了嵌入式系统底层设备接口、处理器子系统，以及总线结构。

并以Intel的ATOM处理器为例，介绍了高性能嵌入式系统的架构设计特点；在软件方面分别介绍了嵌入式系统编程模式、软件优化技术、嵌入式操作系统原理、驱动程序原理，以及基于MeeGo的用户界面编程技术。

《嵌入式系统原理及开发——基于ATOM处理器及Linux实现》面向电子信息及计算机专业本科生嵌入式系统课程教学，也可供于嵌入式领域技术人员参考。

## &lt;&lt;嵌入式系统原理及开发&gt;&gt;

## 书籍目录

- 第1章 嵌入式系统概述
  - 1.1 嵌入式系统的定义
  - 1.2 嵌入式系统的发展历史
  - 1.3 嵌入式系统的特征
- 第2章 嵌入式系统架构及硬件组成
  - 2.1 嵌入式硬件组成
  - 2.2 嵌入式处理器
  - 2.3 Intel嵌入式处理器
  - 2.4 内存子系统
    - 2.4.1 存储单元基本结构和分类
    - 2.4.2 存储器的外部接口
  - 2.5 IO外围设备设备接口
    - 2.5.1 RS232接口
    - 2.5.2 触摸屏接口
    - 2.5.3 显示设备接口
    - 2.5.4 串行总线及通信接口
    - 2.5.5 存储接口
    - 2.5.6 SATA接口硬盘
    - 2.5.7 CF卡
  - 2.6 总线接口
    - 2.6.1 静态存储器总线
    - 2.6.2 PCI04/ISA总线
    - 2.6.3 PCI/PCI04+总线
    - 2.6.4 PCIe总线
- 第3章 嵌入式软件构成与开发方法概述
  - 3.1 嵌入式系统软件特性
  - 3.2 软件开发到运行的流程
  - 3.3 软件编译
  - 3.4 软件启动
  - 3.5 嵌入式软件运行模式
  - 3.6 软件库
  - 3.7 程序固化
  - 3.8 软件调试
  - 3.9 软件开发工具
    - 3.9.1 GNU交叉编译器的生成
    - 3.9.2 Intel的软件开发工具
  - 3.10 嵌入式系统编程模式
    - 3.10.1 多任务程序的实现
    - 3.10.2 嵌入式操作系统
  - 3.11 嵌入式系统的软件可靠性
    - 3.11.1 检查点技术
    - 3.11.2 双进程技术
    - 3.11.3 N-版本软件技术
    - 3.11.4 前向纠错技术(Forward error recovery)
  - 3.12 嵌入式系统的设计方法

## &lt;&lt;嵌入式系统原理及开发&gt;&gt;

## 第4章 嵌入式软件开发工具及软件

## 优化

## 4.1 Intel C++编译器

## 4.2 代码优化方法学及具体技术

## 4.2.1 代码优化方法学

## 4.2.2 基于编译选项的静态优化技术

## 4.2.3 基于源代码结构改进的优化技术

## 4.2.4 基于运行分析的运行时刻代码优化技术

## 4.2.5 基于运行库和编译器库的程序优化

## 4.3 基于TBB的并行软件优化开发

## 第5章 系统启动

## 5.1 引言

## 5.2 BIOS概述

## 5.2.1 BIOS的发展

## 5.2.2 BIOS的功能

## 5.3 基于传统BIOS的系统启动

## 5.4 UEFI及系统启动

## 5.4.1 传统BIOS的缺陷

## 5.4.2 传统BIOS到UEFI、I/EFI的转变

## 5.4.3 UEFI/EFIBIOS的基本架构

## 5.4.4 基于UEFI的系统启动

## 5.5 BootLoader实例分析

## 5.5.1 Grub

## 5.5.2 eLilo

## 5.5.3 Uboot

## 第6章 嵌入式Linux操作系统

## 6.1 嵌入式Linux简介

## 6.2 嵌入式Linux内核配置

## 6.3 嵌入式Linux裁剪

## 6.4 设备驱动

## 6.4.1 设备驱动的作用

## 6.4.2 设备驱动程序分类

## 6.4.3 字符型设备驱动程序的基本构成

## 6.4.4 字符型设备驱动程序的高级功能构成

## 第7章 嵌入式系统低功耗技术

## 7.1 嵌入式系统的低功耗设计

## 7.2 系统节电原理

## 7.3 动态电源管理(DPM)

## 7.4 ACPI(高级配置与电源接口)

## 7.5 其他省电技巧

## 7.6 一些相关工具

## 第8章 嵌入式Linux快速启动技术

## 8.1 系统的启动过程

## 8.2 BootGraph和Bootchart

## 8.3 系统的快速启动技术

## 第9章 嵌入系统的用户界面技术

## 9.1 界面及用户交互程序调试技术

## <<嵌入式系统原理及开发>>

### 9.2 基于Clutter的嵌入式图形界面技术

#### 9.2.1 Clutter编程模型

#### 9.2.2 Clutter场景

#### 9.2.3 演员的属性和静态动作

#### 9.2.4 动态效果

#### 9.2.5 事件处理

#### 9.2.6 Clutter的编程环境

### 9.3 基于MeeGo的图形界面技术

#### 9.3.1 概述

#### 9.3.2 MeeGo软件开发模式

#### 9.3.3 应用程序编程技术

### 9.4 中间层软件编程技术

#### 9.4.1 D-BUS

#### 9.4.2 底层系统层软件编程技术

### 参考文献

## &lt;&lt;嵌入式系统原理及开发&gt;&gt;

## 章节摘录

版权页：插图：4.2.1代码优化方法学 1.程序优化问题的分类 程序优化首先需要明确目标，这依赖于程序运行的硬件环境，图4.5给出了两类的嵌入式处理器运行硬件环境。

图(a)对应的嵌入式系统的程序运行主要依赖于大容量RAM(通常是SDRAM)，而板上ROM仅存放CPU启动初期运行所需的少量代码，ROM内代码的主要功能是从外置存储器把主程序代码和数据搬运到RAM，一旦搬运结束，就让CPU转跳到RAM地址执行主程序。

对于x86架构的嵌入式处理器而言，ROM代码就是BIOS，而外置存储器一般是硬盘或者固态硬盘(SSD，利用FlashROM芯片搭建)。

这类系统的例子包括多媒体机顶盒、视频游戏机等。

它们的程序优化一般集中在两个方面——运行时间最短、代码尺寸最小。

图(b)对应的嵌入式处理器内置有ROM和RAM，分别存放指令和可变数据，这一类系统一般用于控制领域，通常对CPU主频、内存、功耗有严格限制，需要优化的目标更多。

这类系统的例子包括测量仪表、手持POS机等。

它们的优化目标一般包括——运行时刻使用的RAM最小、在限定时间内完成任务消耗的总电能最小、以及在给定的响应时间内完成任务等。

针对这两类硬件环境，优化的总体目标不同，处理方法也各有侧重点。

优化工作需要考虑的第二个问题是明确优化的大致方向策略。

大多数优化工作和速度有关，我们可以通过程序功能大致确定运行程序时主要的时间消耗在哪里。

通常可以分成三种情况：(1)计算为主的代码比如科学计算、信号处理等程序，数据计算占据了主导地位。

对于这一类程序的优化侧重于应用程序级和处理器微架构级。

通过数据计算的简化(如一些运算通过查表得到)运算的并行化(如使用SIMD指令)，以及和Cache相关的数据访问优化获得性能的提升。

(2)IO为主的代码这一类代码运行过程中有大量的时间消耗在外部IO设备的等待过程中，比如数据库操作和网络通信操作，为了提升性能，需要从系统级和应用程序级进行优化，可以使用的优化手段包括利用输入输出缓冲减少零碎的数据通信，使用更紧凑的数据结构和协议降低通信数据量等。现实中也存在着大量介于上面两者之间的程序，比如网络视频压缩传输程序等，这类程序的优化需要从各个层面进行。

(3)解决优化问题的不同层面 程序性能取决于多个方面，包括：代码结构、算法、编译器，以及处理器架构。

面对程序的性能问题如何开展优化？

优化工作从三个层面进行——系统级、应用程序级和处理器微架构级。

针对三个层次优化的顺序和可能的性能提升如图4.6所示。

这三个不同层次的简介、优化对应的优化目标、对象如表4.3所示。

2.软件优化技术的分类 从优化技术所采取的手段来看，软件优化技术分为以下几类：(1)基于编译选项的静态优化技术 该技术基于编译和链接工具提供的命令行选项，选择可执行代码的生成方式，适应程序和处理器硬件的特点，实现程序运行效率和性能的优化。

(2)基于源代码结构改进的优化技术 这一技术通过程序员对代码的主动思考，分析可能存在的性能瓶颈进行优化。

由于能够从较高层次识别软件性能和效率的缺陷，通常能够获得的性能提升也较多。

但这一技术要求程序要有丰富的优化经验和编程实践。

(3)基于软件动态运行分析的“反馈式”优化技术 这一模式的优化技术考虑到代码优化的策略依赖于程序动态运行情况。

比如对视频解码程序，需要通过寻找运行时间最长、运行次数最频繁的函数来进行有针对性的优化。

而这些运行时刻的数据只有通过程序的试运行才能够发现。

因此选用的优化流程通常是：“预编译—试运行，性能评估—重编译并进行优化”，利用预编译的代

## <<嵌入式系统原理及开发>>

码试运行结果反馈给编译器，调整编译过程中代码生成方式，实现代码优化。  
根据程序性能的改进量和优化程度的要求，往往会通过迭代循环的形式进行优化。

<<嵌入式系统原理及开发>>

编辑推荐



<<嵌入式系统原理及开发>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>