

<<嵌入式软件基础>>

图书基本信息

书名：<<嵌入式软件基础>>

13位ISBN编号：9787040161052

10位ISBN编号：7040161052

出版时间：2005-5

出版时间：高等教育出版社

作者：刘易斯

页数：235

字数：340000

译者：陈宗斌

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>

<<嵌入式软件基础>>

前言

你的家中有多少台计算机？
大多数人可能会回答有两台或三台。
你家中有多少个微处理器？
在你回答前仔细考虑一下（提示：不止两三个！
事实上，甚至会有10多个或20多个！
）。

今天，微处理器已嵌入到几乎所有你能想到的以及许多你可能想不到的电器中。它们已变得非常普及——不仅在我们家中，而且在我们的工作场所、我们的汽车、飞机、停车灯、超市、手机中——总之，在我们生活的几乎所有方面。

嵌入式系统为学生表达他们的创造力提供了一个令人兴奋的机会，他们梦想设计出下一代新的小器具，迎合大众的需要。

作为教育工作者，我们的挑战是抓住这个激动人心的机会，并弓！

导年轻人的精神活力，以激励他们掌握这个主题。

目标 本书的最终目标是打下一个基础，为学生掌握多线程编程风格和嵌入式软件的高可靠性要求提供支持。

在本教材中，我们建立了以下目标： 1.理解如何在机器级表示数据，认识到这些表示的后果和局限。

2.掌握嵌入式系统最常用的语言特有的特性，如位操纵和变量访问。

3.了解程序员如何看待处理器体系结构以及如何在汇编级编程，有时是必要的或合适的。

4.了解多种不同风格的I/O编程，并且最终了解如何通过事件驱动方法把数据处理隔离到许多独立的计算线程中。

5.了解非抢先式和抢先式多线程编程、共享资源和临界区，以及如何使用调度来管理系统响应时间。

6.复习作用域、参数传递、递归和内存分配这类主题，以强化基本的编程技能。

7.了解与共享内存对象有关的问题、内存分配如何影响共享内存以及可以使用哪些编程实践来最小化共享内存的发生。

目标读者 本书打算用作计算机科学、计算机工程或电子工程专业课程二年级教材，用来代替计算机组织和汇编语言编程的传统教材。

本书介绍了实践中最常用的汇编方法——以实现小型、快速或专用例程供主程序调用，这些主程序是用诸如C之类的高级语言编写的。

因此，本书仅从“须知”的观点出发来介绍处理器组织和汇编语言，而不是将其作为一个主要目标来讲解。

这种方法为用本书讲授嵌入式软件环境中的汇编语言提供了时间保证。

<<嵌入式软件基础>>

内容概要

本书是对高等院校本科二年级计算机组成原理与汇编语言程序设计的传统教材的全新替代版。

本书以实践中最常运用的方式讲解汇编语言——实现小型、快速或特殊目的的例程，这些例程由主程序（高级语言编写，如C）调用。

通过运用嵌入式软件环境，本书介绍多线程程序设计、可抢占式系统与非可抢占式系统、共享资源和调度，从而为操作系统、实时系统、计算机网络及基于多处理器的设计等后续课程提供了坚实的基础。

本书适用于高等院校工科各专业本科嵌入式计算机系统程序设计、C语言程序设计及汇编语言程序设计类课程，也可供相关技术人员学习参考。

<<嵌入式软件基础>>

书籍目录

序言第1章 导论 1.1 什么是嵌入式系统 1.2 嵌入式软件设计目标有何独特之处 1.3 “实时”意味着什么 1.4 “多任务”意味着什么 1.5 嵌入式处理器的功能有多强 1.6 使用哪些编程语言 1.7 什么是“实时内核” 1.8 如何构建独特的嵌入式应用程序 1.9 典型的嵌入式程序有多大 1.10 本书中使用的软件 习题

第2章 数据表示 2.1 固定精度的二进制数字 2.1.1 按位记数制 2.1.2 二进制—十进制转换 2.1.3 十进制—二进制转换 2.1.4 计数 2.1.5 固定精度和翻转 2.1.6 十六进制表示 2.2 整数的二进制表示 2.2.1 带符号整数 2.2.2 同一个值的正的表示和负的表示 2.2.3 解释2的补码数的值 2.2.4 关于范围和溢出的进一步说明 2.2.5 2的补码和硬件复杂性 2.3 实数的二进制表示 2.3.1 定点表示 2.3.2 使用通用的16.16格式的定点表示 2.3.3 使用通用的32.32格式的定点表示 2.3.4 浮点表示 2.4 文本的ASCII表示 2.5 二进制编码的十进制表示 习题

第3章 充分利用C语言 3.1 整型数据类型 3.2 混合数据类型 3.3 有用的typedef和define 3.4 操纵内存中的位 3.4.1 测试位 3.4.2 设置、清除和反转位 3.4.3 提取位 3.4.4 插入位 3.5 操纵：I/O端口中的位 3.5.1 只写I/O端口 3.5.2 通过读/写区分的端口 3.5.3 通过顺序访问区分的端口 3.5.4 通过写入数据中的位区分的端口 3.6 访问内存映射的I/O设备 3.6.1 通过指针访问数据 3.6.2 数组、指针和“取地址”运算符 3.7 结构 3.7.1 打包的结构 3.7.2 位域 3.8 变型访问 3.8.1 强制转换对象的地址 3.8.2 使用共用体 习题

第4章 程序员眼中的计算机结构 4.1 内存 4.2 中央处理器 4.2.1 运算器 4.2.2 其他寄存器 4.2.3 控制器 4.3 输入/输出 4.4 Intel架构介绍 4.4.1 指令格式 4.4.2 指令操作数 4.4.3 操作数限制 4.4.4 寄存器 4.4.5 栈 4.5 Intel实模式架构 4.5.1 分段寻址 4.5.2 寻址模式 4.6 Intel保护模式架构 4.6.1 段寄存器和全局描述符表 4.6.2 平坦内存模型 4.6.3 寻址模式 4.7 操作数与地址长度覆盖前缀 4.8 Intel数据操纵指令 4.8.1 数据移动、栈和I/O指令 4.8.2 算术指令 4.8.3 按位指令 4.8.4 移位指令 习题

第5章 c语言与汇编语言的融合第6章 输入/输出编程第7章 并发软件第8章 调度第9章 内存管理第10章 共享内存第11章 系统初始化附录A配书光盘上的内容附录BDJGPPC / C++编译器附录CNASM ; I-编译器附录D编程项目附录Elibepc库附录F引导加载程序附录G词汇表

<<嵌入式软件基础>>

章节摘录

当你想运行一个桌面应用程序时，就会通过操作系统的一部分（称为加载程序），把它的可执行映像从磁盘上加载到内存中。

操作系统自身已经在内存中，它是在引导过程中加载到其中的”。

桌面系统用于运行许多不同的应用程序。

因而，会使用读/写主内存，以使完全不同的应用程序能够快速、容易地加载到内存中，无论何时都可根据需要来替换先前的应用程序。

与通用的桌面系统不同的是，嵌入式系统被设计成为服务于单一目的。

如果嵌入式软件已位于内存中，通常没有理由改变它。

这使得使用价格低廉的只读存储器来永久性地存储该程序成为可能。

由于没有必要把程序存储到磁盘上，因此可以显著削减软件数量，而这些软件可能是支持文件系统所必要的。

嵌入式应用的软件和实时内核都作为单一的程序映像存储在只读存储器中。

由于不需要文件系统，因而其内核比常规的操作系统要小得多，并且能够很容易地连接到作为另一种对象文件的程序中。

<<嵌入式软件基础>>

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>